OhMG: Zero-shot Open-vocabulary Human Motion Generation

Anonymous authors
Paper under double-blind review

Abstract

Generating motion in line with text has attracted increasing attention nowadays. However, open-vocabulary human motion generation still remains touchless and undergoes the lack of diverse labeled data. The good news is that, recent studies of large foundation models (e.g., CLIP) have demonstrated superior performance on few/zero-shot image-text alignment, largely reducing the need for manually labeled data. In this paper, we take the advantage of CLIP for open-vocabulary 3D human motion generation in a zero-shot manner. Specifically, our model is composed of two stages, i.e., text2pose and pose2motion. For text2pose, to address the difficulty of optimization with direct supervision from CLIP, we propose to carve the versatile CLIP model into a slimmer but more specific model for aligning 3D poses and texts, via a novel pipeline distillation strategy. Optimizing with the distilled 3D pose-text model, we manage to concretize the text-pose knowledge of CLIP into a text2pose generator effectively and efficiently. As for pose2motion, drawing the inspiration of the advanced language model, we pretrain a transformer-based motion model, which makes up for the lack of motion dynamics of CLIP. After that, by formulating the generated poses from the text2pose stage as prompts, the motion generator can generate motions referring to the poses in a controllable and flexible manner. The code will be released here.

1 Introduction

Motion generation has attracted increasing attention due to its practical value in the fields of virtual reality, video games and movies. As the motion capture techniques become mature, motion data can be collected with fewer human efforts (Mahmood et al., 2019). However, when it comes to labeling the collected motions, the diversity of textual descriptions is usually affected by the labeling instructions and the backgrounds of the crowd annotators, which might introduce unexpected selection biases (Pearl & Mackenzie, 2018) and limit its generality. The scarcity of diverse textual descriptions for different motions is one of the major obstacles to open-vocabulary motion generation. Nevertheless, the recent works on large-scale multi-model foundation models, e.g., CLIP (Radford et al., 2021b) or ALIGN (Jia et al., 2021), have shown the surprising capability to align diverse text and images in a few/zero-shot manner and largely reduce the need for manually labeling. Equipped with the foundation models, classical methods for text-to-image generation (Gal et al., 2021; Frans et al., 2021; Ramesh et al., 2022) and robotic vision grasping (Shridhar et al., 2022), are able to generalize to unseen textual descriptions or objects during inference. However, these methods stay at the level of using the feature representation ability of the foundation model. In this work, we take a step further and concretize the knowledge of the well-known foundation model, i.e., CLIP, to facilitate the zero-shot open-vocabulary 3D human motion generation.

The foundation model, CLIP (Radford et al., 2021b), is a language-image pretrained model for aligning images and texts. However, since it only trained with static images, it implies that CLIP lacks the knowledge of motion dynamics for motion generation and can only be used for static pose generation. To this end, it's reasonable to leverage CLIP for static pose generation and then combine it with the motion dynamics learned from the collected motion data to generate the complete motion (Hong et al., 2022b). In this paper, we adopt a two-stage text2motion generation model for zero-shot **O**pen-vocabulary **h**uman **M**otion **G**eneration, termed as OhMG. OhMG consists of two stages, i.e., text2pose and pose2motion stages. Briefly, text2pose translates a textual motion description into the signature pose of the motion, which is then served as a condition input to the

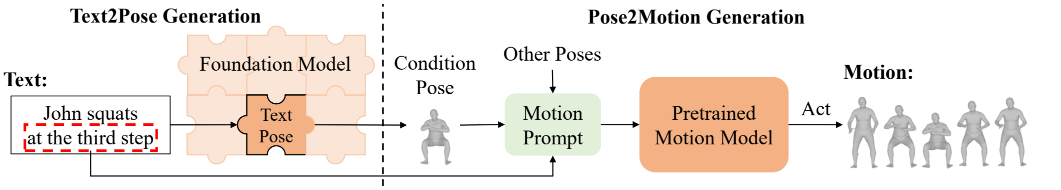

Figure 1: An overall sketch of OhMG. A text is fed to the text2pose generator to obtain a signature pose at the text2pose stage. Then, the pose can combined with other information, e.g., the text and the other poses, to create a motion prompt for the pretrained motion model to generate a motion.

pose2motion stage to obtain the complete motion. The overall sketch of OhMG is illustrated in Fig. 1. In the following, we further elaborate on each stage of our method.

At the text2pose stage, a text2pose generator learns to generate text-consistent 3D poses under the supervision of CLIP. However, as mentioned above, CLIP is not pretrained for aligning texts and 3D poses. To make use of CLIP, a promising workaround is to project the 3D human model to 2D images and then apply CLIP to measure the alignment between the images and the input text (Hong et al., 2022b). When the whole pipeline is implemented in a differentiable manner, the input pose of the pipeline should be able to adjust and fit the input text via gradient descent. Unfortunately, in practice, we found it difficult to optimize the poses with this pipeline directly. The potential reason could be that, by using CLIP, there could be various images to present the same text (Ramesh et al., 2022; Zhou et al., 2022). And the rendered 2D images of the 3D human model might not be the optimal ones. In this case, optimization with CLIP might adjust the 3D pose in an unexpected direction. Furthermore, the pipeline is complex and requires substantial computing resources, which also hampers the optimization process. To address these problems, we propose a novel knowledge distillation method, termed pipeline distillation, which learns an end-to-end neural network to replace the complex pipeline. By learning from substantial poses and their output of the pipeline, pipeline distillation carves out a smaller but more specific model of CLIP for aligning 3D poses and texts. In our experiments, by using the distilled model, optimization becomes significantly efficient and we manage to learn a text2pose generator for generating open-vocabulary poses. Interestingly, we found that the text2pose generator can be trained without any real-world text or pose explicitly.

As for the stage of pose2motion, a motion generator is required to synthesize the motion containing the given/condition poses. Previous work (Hong et al., 2022b) adopts the decoder of a pretrained motion VAE and searches in its latent space for preferable motion. The latent code is updated via gradient descent to minimize the distance between the poses of the decoded motion and the condition poses. However, this method requires iterative updates during deployment, which is time-consuming. Further, since the latent space of the motion, VAE is high-dimensional and not necessarily convex, the generated motion is difficult to optimize. Differently, we view the relationship between poses and motion as the relationship between words and sentences in the field of natural language processing. And we pretrain a motion model via mask-and-reconstruction self-supervised learning as used in the advanced language model (Devlin et al., 2018). And during inference, the condition poses can be treated as the unmasked poses to prompt the motion model (Brown et al., 2020; Han et al., 2021) to synthesize the rest of the poses, resulting in a complete motion. We find that the motion model is easy and flexible to control to generate diverse motions for the given poses.

Overall, the contributions of our OhMG are as follows. **For text2pose stage:** 1) We propose a novel text2pose generator that mines the knowledge from the foundation model, i.e., CLIP. 2) To overcome the difficulty of optimization, we propose a pipeline distillation strategy that turns the complex pipeline into a slimmer model for aligning 3D poses and texts. **As for pose2motion stage:** 3) We consider the motion generation as the same as the language modeling and pretrain a motion model via mask-and-reconstruction self-supervised learning. 4) Inspired by prompt design for probing knowledge from pretrained model (Han et al., 2021), we reformulate the condition poses as prompts to leverage the motion model to generate motion in a controllable and flexible manner.

2 Related Work

Conditional Motion Generation. There are several large-scale motion capture datasets (Cai et al., 2022; 2021; Ionescu et al., 2014; Mahmood et al., 2019; Mehta et al., 2016; Varol et al., 2017; von Marcard et al., 2018), which are in forms of 3D keypoints or parameters of 3D human model

SMPL (Pishchulin et al., 2017). For conditional motion generation, some works (Aggarwal & Parikh, 2021) focus on music-conditioned motion synthesis. While DVGANs (Lin & Amer, 2018), Text2Action (Ahn et al., 2018) and Language2Pose (Ahuja & Morency, 2019) generate motions conditioned on short texts using fully annotated data. Action2Motion (Guo et al., 2020) and Actor (Petrovich et al., 2021) condition the motion generation on pre-selected action classes. However, these methods require large amounts of data (Hong et al., 2022a) with annotations of action classes or language descriptions, which limits their applications.

Zero-shot Text-driven Generation. The ability to zero-shot generalize to unseen categories is first shown by (Reed et al., 2016). CLIP and DALL-E (Radford et al., 2021b) further show the incredible text-to-image synthesis ability by excessively scale-up the size of training data. Benefiting from the zero-shot ability of CLIP, many amazing zero-shot text-driven applications (Frans et al., 2021; Patashnik et al., 2021; Peng et al., 2021b) are being developed. Combining CLIP with 3D representations like NeRF or mesh, zero-shot text-driven 3D object generation (Jain et al., 2022; Jetchev, 2022; Michel et al., 2022; Sanghi et al., 2021) and manipulation (Peng et al., 2021a) have also come true in recent months.

CLIP aided Methods. Neural networks have successfully learned powerful latent representations coupling natural images with natural language describing it (He & Peng, 2017; Radford et al., 2021a). A recent example is CLIP (Radford et al., 2021a), a model coupling images and text in deep latent space using a constructive objective (Hadsell et al., 2006). By training over a hundred million images and their captions, CLIP gained a reach semantic latent representation for visual content. This expressive representation enables high-quality image generation and editing, controlled by natural language (Gal et al., 2021; Frans et al., 2021). Even more so, this model has shown that connecting the visual and textual worlds also benefits purely visual tasks (Vinker et al., 2022), simply by providing a well-behaved, semantically structured, latent space.

Closer to our method are works that utilize the richness of CLIP outside the imagery domain. In the 3D domain, CLIP's latent space provides a useful objective that enables semantic manipulation (Sanghi et al., 2021; Michel et al., 2022) where the domain gap is closed by a neural rendering. CLIP is even adopted in temporal domains (Luo et al., 2021; Fang et al., 2021) that utilize large datasets of video sequences that are paired with text and audio. MotionCLIP (Tevet et al., 2022) takes the advantage of the power representation ability of CLIP and utilizes a limited amount of human motion sequences that are paired with text to learn a text2motion generator. Most related to us, AvatarCLIP is the first work for zero-shot open-vocabulary human motion generation. Unlike ours, their work still relies on the representation of CLIP for matching and fails to use CLIP for optimization.

3 Preliminaries

In this paper, we investigate the zero-shot open-vocabulary human motion generation by mining the text-pose knowledge from the CLIP (Radford et al., 2021b). Specifically, the most important tool used in our method is CLIP there are several key concepts/tools used in our method, including the foundation model CLIP. In the following, we briefly introduce the task and the frequently-used notations as well as the CLIP model.

Task description and notations. Open-vocabulary 3D human motion generation takes in a natural language motion description d (for example, "Amy is shooting a basketball") and searches for a motion m in line with d. A motion is a sequence of 3D poses, $m = [p_t]_{t=1:T}$, where p is the 3D pose, t stands for the timestep and T is the maximum length of a motion. The representation of 3D pose p can be in different formats, e.g., an axis-angle representation $p^a \in \mathbb{R}^{J\times 3}$, an 6D-rotation representation $p^r \in \mathbb{R}^{J\times 6}$ or an latent representation $p^l \in \mathbb{R}^{32}$ of VPoser (Pavlakos et al., 2019). Here, J denotes the number of used joints in the human model and VPoser is a popular pretrained pose VAE model. Any of these representations can be used to generate 3D human meshes. In this paper, we use a popular parametric human model, SMPL, for its strong interpretability and compatibility with modern graphics platforms. SMPL (Loper et al., 2015) is a parametric human model driven by large-scale aligned human surface scans (Pishchulin et al., 2017). We can feed the pose parameters to SMPL to obtain meshes v, denoted as $v = \mathcal{M}_{\text{SMPL}}(p)$. Meshes v are represented by a set of 3D positions of the vertices. Note that, SMPL model also requires other input, such as faces and body shapes, which are set as default and ignored for brevity in our paper.

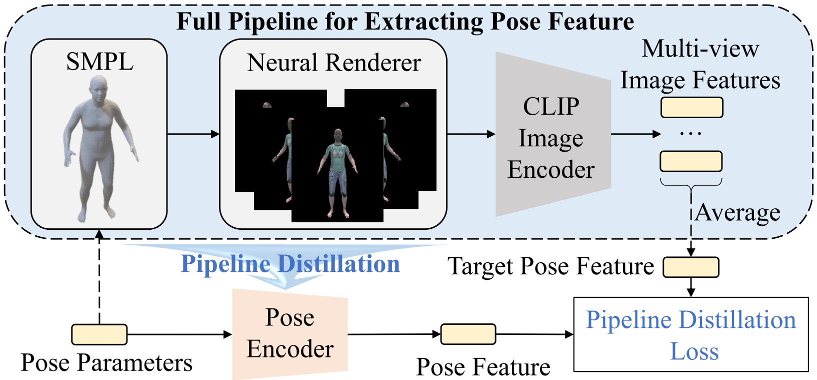

Figure 2: Pipeline distillation. The upper part is the original pipeline for extracting pose feature for alignment. Our pipeline distillation adopt a end-to-end neural network, i.e., pose encoder, which takes in the same input and learns to predict the output of the pipeline.

CLIP. CLIP (Radford et al., 2021b) is a vision-language pre-trained model trained with large-scale image-text datasets. It consists of an image encoder E_o and a text encoder E_d. Here, we use o to denote image and d to represent text. The encoders are trained in the way that the latent codes of paired images and texts are pulled together and unpaired ones have pushed apart. Formally, the CLIP loss function is defined as

$$\mathcal{L}_{\text{CLIP}}(\mathbf{o}, \mathbf{d}) = -\sum_{i=1:B} \log \mathbf{Pr}(o_i|d_i) - \sum_{i=1:B} \log \mathbf{Pr}(d_i|o_i), \tag{1}$$

where $\mathbf{o}$ and $\mathbf{d}$ is the sets of images $\{o_i\}_{i=1:B}$ and texts $\{d_i\}_{i=1:B}$, and B is the batch size. $\mathbf{Pr}$ is the softmax probability of the o_i given d_i in a batch, vice versus. Particularly, to calculate $\mathbf{Pr}(o_j|d_i)$, the cosine similarity between text feature $f_{o_i} = E_o(o_i)$ and each image feature $f_{d_j} = E_d(d_j)$ of the batch data are calculated, and the temperature-softmax operation is applied to the cosine similarities. Formally, for calculating $\mathbf{Pr}(o_j|d_i)$:

$$\text{cossim}(f_i, f_j) = \frac{f_i^T f_j}{|f_i||f_j|}, \quad \mathbf{Pr}(o_i|d_i) = \frac{\exp\left(\text{cossim}(f_{o_i}, f_{d_i})/H\right)}{\sum \exp\left(\text{cossim}(f_{o_j}, f_{d_i})\right)/H}, \tag{2}$$

where H is the temperature to adjust the sensitivity of softmax. By optimizing Equ. (1), CLIP learns a joint latent space where images and texts are well-aligned relatively. For convenience, we use **CLIP score** to stand for the cosine similarity between text and image features from CLIP.

4 OhMG: Open-vocabulary Human Motion Generation

As shown in Fig. 1, our method includes two stages, i.e., text2pose and pose2motion. For the first stage, we probe the versatile CLIP to distill its text-pose knowledge. In this stage, we found that it's difficult to adjust the pose with the original CLIP via gradient descent. And we propose a novel pipeline distillation to address the problem. For pose2motion, we draw the inspiration from advance nature language model pretraining (Devlin et al., 2018). We pretrain a transformer-based (Petrovich et al., 2021; Vaswani et al., 2017) motion model by mask-and-reconstruction self-supervised learning. And for generating motion in a controllable and flexible manner, we reformulate the condition poses from text2pose stage as prompt (Brown et al., 2020) to the pretrained motion model. In the following, we elaborate on each part in detail.

4.1 Text2Pose

To generate a pose given text, it requires a multi-model feature space where the features of texts and poses are well aligned. However, there lacks a multi-model pretrained models for aligning 3D poses and texts. A promising workaround is to render the 3D pose into multi-view images and then use CLIP to measure the alignment between images and text. The illustration of this process is in the upper part of Fig. 2.To address the difficulty of optimization using the complex pipeline as well as reducing the computation costs, we propose a a novel pipeline distillation strategy to learn an

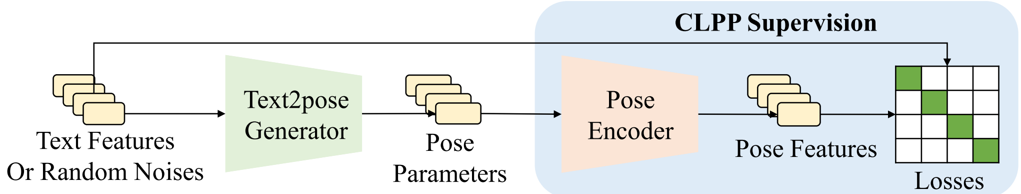

Figure 3: Training process of text2pose generator. During training, either text features extracted from real-world texts or random noises can be used as input features. The text2pose generator takes in the input features and predicts the pose parameters supervised by CLPP.

efficient and effective pretrained model for 3D pose and text. In the following, we describe how pipeline distillation is conducted. After that, we elaborate how to use the pretrained model for 3D pose and text to learn a text2pose generator.

Pipeline Distillation. As discussed above, it's non-trivial to conduct gradient descent for optimizing the pose with the original complex pipeline. In this part, we propose the simplified complicated pipeline into a simple end-to-end neural network. In other words, we train a multimodel pretrained model for 3D poses and text based on existing model for images and texts. To our best knowledge, this work is the first to conduct distillation strategy to address the optimization problem, and it is also the first work to pretrain a model for aligning 3D pose and text. To distinguish from CLIP, we name our distilled model for 3D pose and text as CLPP which stands for Contrastive Language-Pose Pretraining. Note that, CLPP still use the original text encoder E_d of CLIP but replacing the image encoder E_o with pose encoder E_p.

To train CLPP, we samples poses from the AMASS dataset (Mahmood et al., 2019), and the poses are processed through the above-mentioned pipeline to obtain the final pose features, denoted as f_p^*. After that, the poses and their pose features are taken as the inputs and targets for training the pose encoder, i.e., E_p. The pipeline distillation loss is formulated as :

$$\mathcal{L}_{E_p}(p, f_p^*) = ||E_p(p) - f_p^*||_2 - \text{cossim}\big(E_p(p), f_p^*\big), \tag{3}$$

where the first term of Equ.(3) is for reducing the element-level distance between the predicted feature and the target feature. While the second term of Equ.(3) is to reduce the angular difference between the features. The overall training sketch is shown in Fig. 2.

Generalized Text2Pose Generator. Unlike the prior work which conducts optimization/matching after having a text (Hong et al., 2022b), our text2pose generator follows the conventional deep learning method that trained generator can handle various text requests. Moreover, benefit from low computation cost with CLPP, we manage to optimize the generator using a similar loss as CLIP loss (Equ.(1)) which requires a large batch size (Radford et al., 2021b). Since we replace CLIP with CLPP, we use $\mathcal{L}_{\text{CLPP}}$ instead. And we found that using $\mathcal{L}_{\text{CLPP}}$ is much more stable than maximizing the CLIP score between text and pose. Formally, $\mathbf{f_d}$ denotes a batch of text features $f_{d_1}, ..., f_{d_B}$ extracted using E_d, and G_{t2p} represents the text2pose generator. The loss function is:

$$\mathcal{L}_{\text{t2p}}(G_{\text{t2p}}(\mathbf{f_d}), \mathbf{f_d}) = \mathcal{L}_{\text{CLPP}}\Big(E_p\big(G_{\text{t2p}}(\mathbf{f_d}), \mathbf{f_d}\big)\Big) + ||G_{\text{t2p}}(\mathbf{f_d})||, \tag{4}$$

where . The second term of the loss function is used to regulate the predicted latent pose to close to the prior distribution of VPoser. However, from Equ.(4), we need to obtain various $\mathbf{d}$ to extract $\mathbf{f_d}$ for training. Fortunately, we can easily collect substantial motion descriptions on the internet. Furthermore, it occurs to us that we can also randomly sample $\mathbf{f_d}$ from a random distribution, e.g. uniform distribution or Gaussian distribution. In our experiment, we found that by training with random sampled $\mathbf{f_d}$, the text2pose generator can be generalized to the real text and obtain surprisingly good performance. This observation reveals the potential of our method to distill other kinds of knowledge from CLIP without training textual descriptions.

4.2 Pose2Motion

In our paper, we consider the motion similar to a language sentence and learn a pretrained motion model and reformulate the condition pose as a prompt of the motion model to generate motion in a controllable and flexible manner.

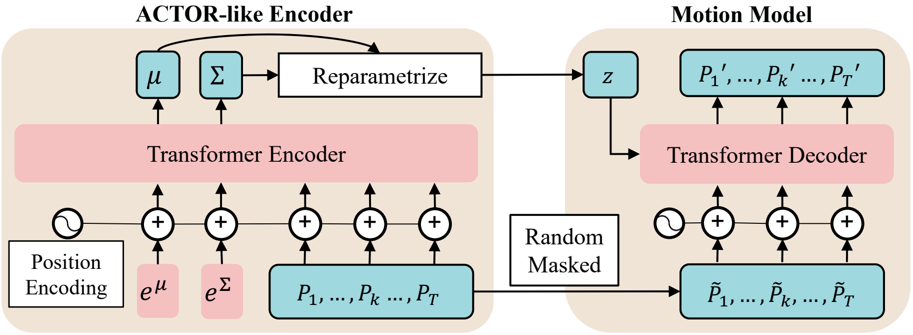

Figure 4: Training process of motion model. On the left is an ACTOR-like Encoder that takes in original motion and extracts the latent code. The right part is a conditional Motion VAE decoder which takes in randomly masked input and the latent code to reconstruct the original motion.

Motion Model Pretraining. For pretraining a motion model, we draw the advanced self-supervised learning of language model (Devlin et al., 2018), which randomly masks a certain proportion of input data and learns to reconstruct the masked data. Our motion model also follows a similar training strategy. Specifically, during training, random proportion of poses of a motion $[p_t]_{t=1:T}$ are masked by a learnable embedding $e^{\text{mask}} \in \mathcal{R}^{|p|}$. Formally, the new input $\tilde{p}_t$ is generated by $\tilde{p}_t = c_t \times p_t + (1 - c_t) \times e^{\text{mask}}$, where $c_t \in \{0, 1\}$ is a binary random condition sampled for each timestep $t \in [1, T]$. When c_t is true, the original pose is preserved. Otherwise, the pose is replaced by the mask embedding.

Then a transformer-based motion model is learn to take in $[\tilde{p}_t]_{t=1:T}$ and predict $[p'_t]_{t=1:T}$ to reconstruct the original motion $[p_t]_{t=1:T}$. However, since the proportaion of changed poses is random, it causes the learning process unstable. To this end, we further draw the lessons from the ACTOR (Petrovich et al., 2021) and formulate our pretrained motion model as a conditional VAE. As illustrated in Fig. 4.2, our model includes an ACTOR-like motion encoder and a motion model, where the motion encoder takes in the original motion sequence and predicts the latent for the motion decoder to predict the reconstructed motion. Formally, the loss function is

$$\mathcal{L}_{mm}(p'_t, p_t) = ||p_t - p'_t||_2 + ||\mathcal{M}_{\text{SMPL}}(p_t) - \mathcal{M}_{\text{SMPL}}(p'_t)|| + KL, \tag{5}$$

where KL is the KL-divergence regularization term to pull the predicted latent to the prior normal distribution. After pretraining the motion model, we only use the motion decoder for pose2motion generation. In the following, we reformulate the poses to be similar as $[\tilde{p}_t]_{t=1:T}$ to prompt the motion model to generate motion.

Motion Prompt. Prompt is a newly rising topic (Brown et al., 2020; Sun et al., 2022; Han et al., 2021) for adapting large foundation model to downstream applications (Jia et al., 2022; Chen et al., 2022). The major motivation of prompt is to reformulate the downstream tasks into the form of the training input of the foundation model. By this means, the pretrained foundation model can be directly used for downstream tasks without finetuning. Inspired by this, with the pretrained motion model, we can reformulate the condition posed by different prompt designs to generate motion in a controllable and flexible manner. Formally, a prompt is a sequence of poses $[\tilde{p}_1]_{t=1:T}$ filled with e^{mask}. And we can change the prompt by replacing a pose $\tilde{p}_k$ of the prompt by the condition pose p, i.e. $\tilde{p}_k = p$. As shown in Fig. 1, we can control the generator to synthesize motion containing the given poses at different positions by designing different prompts.

5 Experiments

We first introduce the datasets and baseline methods used in our experiments. Next, we ablate the pose generation, including the CLPP and text2pose generator. After that, we evaluate the performance of the motion generation. For saving the room, we pose the visual results in Appendix, and we strongly recommend the reader to view the Appendix for better understanding.

General Settings. We train our model on the AMASS motion dataset (Mahmood et al., 2019). It unifies 15 different optical marker-based mocap datasets by representing them within a common framework. The dataset contains more than 40 hours of motion data, spanning over 300 subjects,

Table 1: The features difference and inference efficency of CLPP in comprison to original pipeline. The arrow ↑ indicates the performance is better if the value is higher. Vice versus.

	Batch size ↑	Inference speed (sec) ↓	MSE ↓	Cosine Sim. ↑
Original Pipeline	15	1.2068	-	-
Pipeline Distillation	292350	0.0172	4.56e-4	0.9983

and more than 11000 motions. We down-sample the data to 30 frames per second and cut it into sequences of length 60. For real-world texts for training and evaluation, we adapt the BABEL motion description dataset (Punnakkal et al., 2021) by removing lengthy descriptions, resulting in a dataset with the size of 4178. For SMPL model, we input the pose body with global rotation fixed, following the same setting as in AvatarCLIP. Particularly, the checkpoint of CLIP, i.e. ”CLIP-ViT-B/32”, is used for extracting both image features and text features.

Baselines. In the following, we enumerate the related baseline methods for pose and motion generation, respectively. Since this topic is new and there is only one prior work, i.e., AvatarCLIP (Hong et al., 2022b), to the best of our knowledge, we mostly follow their baselines in this paper. To reduce confusion, we use *italic* font for the names of the baselines. To evaluate the performance of text2pose generation, the related baselines are listed in the following. 1) ***Matching*** (Hong et al., 2022b) uses CLIP to match among a set of poses according to the texts. The poses are 4096 cluster poses from AMASS using KMeans. 2) ***Optimize*** (Hong et al., 2022b) optimizes the pose parameters using the complex pipeline described in our paper to maximize the CLIP score. 3) ***VPoserOptimize*** (Hong et al., 2022b) is similar to *Optimize* but optimizes the latent pose of VPoser instead. As for text/pose to motion generation. There are several approaches to achieving that. 1) ***Interpolation*** (Hong et al., 2022b) linearly interpolates each pair of latent poses. 2) ***AvatarCLIP*** (Hong et al., 2022b) directly optimizes in the latent space of a decoder of pretrained motion VAE by pulling the generate motion close to the reference poses. 3) ***MotionCLIP*** (Tevet et al., 2022) is training with labelled data. It trains a motion VAE and pushes the latent space close to both of text and image feature spaces of CLIP. And during inference, *MotionCLIP* uses the text features from CLIP as the latent code and decodes it into a motion.

5.1 TEXT2POSE GENERATION

At the stage of text2pose, there are two modules of interest, i.e., the pipeline distilled model for aligning 3D poses and texts, namely CLPP, and the text2pose generator. As for CLPP, we measure the difference between the predicted features and the actual features from the original pipeline. Besides, we also testify to the efficiency of CLPP. The results are reported in Tab. 1. From the results, we can observe that the CLPP can extract similar features as the original pipeline with a small mean square error and high cosine similarity between the predicted and the ground-truth features. Nevertheless, CLPP can obtain significant improvement in space occupancy and inference speed by about 20,000x and 700x relative improvement on one NVIDIA V100 Tensor Core (32G).

Moreover, we also evaluate the effectiveness of CLPP in comparison to the original pipeline. Before the analysis, we describe the metrics used in this part. As listed in Tab. 2, there are six metrics for measuring **CLIP score** between the predicted poses and the texts, testifying whether the generated pose are within real-world pose distribution (**In-distrib.**), evaluating the diversity of the generated poses (**Diverse**). In-distrib. is the reconstruction error of VPoser and we multiply. And the latter **Top1**, **Top10** and **Top50** stands for the accuracy of matching among all generated poses for all texts. If the matched pose is the generated pose of the text using the baseline method, we say the matching is accurate. Since the evaluation texts in BABAL contains many similar textual descriptions, top1 usually cannot reflect the actual performance of each method. To this end, we also include Top5 and Top10 accuracies.

To evaluate the effectiveness of CLPP, we first show the difficulty of using the original pipeline for optimization. In Tab. 2, we use (I) to stand for the random initialization without optimization. We observe that the CLIP scores of *Optimize*, *VPoserOptimize*, *Optimize* (I) and *VPoserOptimize* (I) are almost the same. And as pointed in Hong et al. (2022b), *Matching* among poses candidates can obtain higher CLIP scores. Ideally, the optimization-based methods should at least achieve a similar or higher CLIP score than direct matching. Thus, these observations imply that the original

Table 2: Comparison among text2pose baselines. The arrow ↑ indicates the performance is better if the value is higher. Vice versus.

	CLIP Score ↑	In-distrib. ↓	Diverse ↓	Top1 ↑	Top10 ↑	Top50 ↑
Matching	0.2615	0.0150	0.2877	0.0119	0.0821	0.2793
Optimize (I)	0.2446	8.5731	0.0416	0.000	0.0041	0.0138
Optimize	0.2468	8.6403	0.0446	0.0000	0.0041	0.0146
VPoserOptimize (I)	0.2441	0.0133	0.0434	0.0000	0.0025	0.0138
VPoserOptimize	0.2436	0.0142	0.0491	0.0000	0.0019	0.0113
Ours (TS)	0.2580	0.5891	0.0522	0.0117	0.0572	0.1606
Ours (TC)	0.2586	0.6243	0.0446	0.0108	0.0677	0.1620
Ours (NC)	**0.2698**	0.1217	**0.0394**	0.0883	0.3396	0.6261
Ours (NTC)	0.2697	0.1261	0.0402	**0.0993**	**0.3562**	0.6130
Ours (NLC)	0.2689	**0.0138**	0.0446	0.0792	0.3394	0.6601
Ours (NTLC)	0.2689	**0.0138**	0.0455	0.0828	0.3454	**0.6857**

pipeline fails to provide direct supervision for optimization. However, with CLPP, our methods can all obtain similar or higher CLIP scores than *Matching* as shown in Tab. 2. For briefness, among our experiments, (T) stands for using real-world texts as training data, (N) stands for using noise features as training data, (S) represents optimization by maximizing CLIP score, (C) represents optimization by minimizing $\mathcal{L}_{\text{CLPP}}$ and (L) means training with L2-norm regularization of Equ.(4).

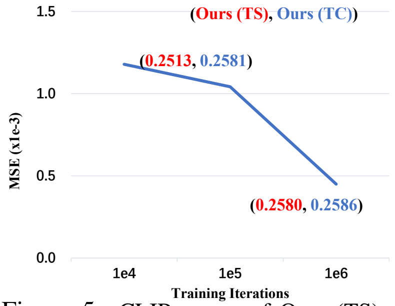

Figure 5: CLIP scores of Ours (TS) and Ours (TC) with CLPP at different iterations.

From the results in Tab. 2, we observe that in comparison to *VPoserOptimize*, Ours (TS) can obtain significant improvement upon CLIP score. It implies that CLPP learns the knowledge from CLIP and can effectively supervise the optimization process. As for optimization by maximizing score, i.e., Ours (TS), or minimizing $\mathcal{L}_{\text{CLPP}}$, i.e., Ours (TC), we find that both manners result in similar performance across different metrics. However, in our experiments, maximizing the score is more sensitive to the convergence of CLPP. As shown in Fig. 5, Ours (TS) degrades severely when CLPP is only trained with 1e5 iterations while Ours (TC) can still achieve stable performance. We contribute this stableness to the denser supervision feedback from not only the positive text-pose pair but also the negative pairs.

Later, we also conduct a series of experiments by testifying the feasibility of training with noise features sampled from Gaussian or Uniform distribution without any real-world texts. Interestingly, we found that Ours (NC) use of only noise features can outperform previous methods with real-world texts significantly on all metrics. It's worth noticing that, the texts used for training are testing are the same as the previous methods with real-world texts. This implies that learning with out-of-distribution random noises can generalize to in-distribution performance and even better. The potential reason could be there is a subtle yet non-ignorable gap between CLPP and the original pipeline. Therefore, learning with a small set of texts might be trapped by the gap. Therefore, training with tremendous random noise can better make use of comprehensive feedback to jump out of the gap. It also explains the source of the stableness of CLPP loss in the last paragraph. And by regularizing the latent code of the predicted poses to close to the prior distribution of VPoser, Ours (NLC) and Ours (NTLC) can predict in-distribution poses similar to the real-world poses of *Matching*.

5.2 Motion Generation

As for motion generation, our method learns a pretrain motion model following language modeling and uses text-consistent poses to prompt the model for synthesizing complete motions. To evaluate the performance of motion generation, we first evaluate the ability to synthesize dynamics-consistent motion with the condition poses. After that, we evaluate the overall performance of text2motion generation.

Table 3: Evaluation for conditional motion generation. The arrow ↑ indicates the performance is better if the value is higher. Vice versus.

	1p↓	2p↓	3p↓	In-distrib.↓	Top1↑	Top10↑	Top50↑
MotionCLIP	-	-	-	0.7943	0.0050	0.0419	0.1649
Interpolation	**0.0900**	**0.0865**	**0.0868**	0.5563	0.0029	0.0184	0.0804
AvatarCLIP	1.9022	2.4353	2.6163	0.3147	0.0014	0.0096	0.0421
Ours	0.6252	0.5219	0.4797	**0.0877**	**0.0689**	**0.2054**	**0.4349**

For evaluating pose2motion, we are interested in whether the given poses exist in the generated motion. To this end, we construct three test sets for three different settings, i.e. 1-pose, 2-pose, and 3-pose, where the prefix number indicates the number of given poses for generating a motion. Among these experiments, we measure the minimal difference between the given poses and the poses of the generated motion, formally,

$$\text{K-pose}(m, \{p_k\}_{j=k:K}) = 100/K \sum_{k=1:K} \min_{p_j \in m} ||p_k - p_j||_2, \tag{6}$$

where m is the generated motion and $\{p_k\}_{j=k:K}$ is the condition poses. For multiple given poses, we calculate have for each pose and use their average results. The results are shown on the left of Tab. 3. From the results, we observe that the *Interpolation* method can obtain the best K-pose results. The reason is straightforward *Interpolation* starts from the given poses and conducts linear interpolation between each of them. The reason why there is still a small error for *Interpolation* method is that interpolation is conducted on the latent space of VPoser. However, interpolation does take motion dynamics into account. Besides, our method can obtain clear improvement upon the strong baseline *AvatarCLIP*. Moreover, our method does not need iterative optimization during deployment. It means that our method can better generate motion according to the given poses.

Finally, we evaluate the overall text2motion generation performance across supervised / zero-shot methods using CLIP. Since all of these methods are CLIP-related, we evaluate whether the motion is distinguished by CLIP by measuring the matching accuracy of all poses of all generated motions. When the matched poses are located within the motion generated according to the text, we say the matching is accurate. And we also calculate the Top1, Top10, and Top50. Moreover, to quantify whether the generated motion follows real-world motion dynamics, we adopt another pretrained motion VAE to calculate the reconstruction error (i.e., In-distrib.) of generated motions for different methods. The smaller the reconstruction error is, the more likely the motion is within the training distribution of the motion VAE. From the results in Tab. 3, we find that among all baselines, our method obtains the best result in terms of In-distrib. as well as TopK accuracies by a clear margin. Worth noticing that, even though our method is not trained with paired data, it can outperform *MotionCLIP* by mining text-pose knowledge from CLIP and using better motion generation architecture. According to the results of In-distrib., we observe that although *Interpolation* is easy to control and preserve condition poses, it does not take motion dynamics into account and results in poor In-distrib performance.

6 Conclusion

In this paper, we propose a zero-shot open-vocabulary human motion generation (OhMG) framework, which leverages the text-pose knowledge from CLIP to build a text2pose generator and use the generated pose to prompt a motion model to generate motion. Extensive experiments have testified that our text2pose generator can learn to generate pose supervised from our distilled CLPP. And the motion generator can generate text-consistent motion by formulating the pose as a prompt to the motion model.

References

Gunjan Aggarwal and Devi Parikh. Dance2music: Automatic dance-driven music generation. *arXiv: Sound*, 2021.

Hyemin Ahn, Timothy Ha, Yunho Choi, Hwiyeon Yoo, and Songhwai Oh. Text2action: Generative adversarial synthesis from language to action. *international conference on robotics and automation*, 2018.

Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. *international conference on 3d vision*, 2019.

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.

Zhongang Cai, Mingyuan Zhang, Jiawei Ren, Chen Wei, Daxuan Ren, Jiatong Li, Zhengyu Lin, Haiyu Zhao, Shuai Yi, Lei Yang, et al. Playing for 3d human recovery. *arXiv preprint arXiv:2110.07588*, 2021.

Zhongang Cai, Daxuan Ren, Ailing Zeng, Zhengyu Lin, Tao Yu, Wenjia Wang, Xiangyu Fan, Yang Gao, Yifan Yu, Liang Pan, et al. Humman: Multi-modal 4d human dataset for versatile sensing and modeling. *arXiv preprint arXiv:2204.13686*, 2022.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, pp. 2778–2788, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv: Computer Vision and Pattern Recognition*, 2021.

Kevin Frans, L. B. Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv: Computer Vision and Pattern Recognition*, 2021.

Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv: Computer Vision and Pattern Recognition*, 2021.

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. *acm multimedia*, 2020.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. *computer vision and pattern recognition*, 2006.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021.

Xiangteng He and Yuxin Peng. Fine-grained image classification via combining vision and language. *computer vision and pattern recognition*, 2017.

Fangzhou Hong, Liang Pan, Zhongang Cai, and Ziwei Liu. Versatile multi-modal pre-training for human-centric perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16156–16166, 2022a.

Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535*, 2022b.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

Ajay Jain, Ben Mildenhall, Jonathan T. Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. 2022.

Nikolay Jetchev. Clipmatrix: Text-controlled creation of 3d textured meshes. 2022.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pp. 4904–4916. PMLR, 2021.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. *arXiv preprint arXiv:2203.12119*, 2022.

Xiao Lin and Mohamed R Amer. Human motion modeling using dvgans. *arXiv preprint arXiv:1804.10652*, 2018.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *international conference on computer graphics and interactive techniques*, 2015.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval. *arXiv: Computer Vision and Pattern Recognition*, 2021.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. Amass: Archive of motion capture as surface shapes. *international conference on computer vision*, 2019.

Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. *international conference on 3d vision*, 2016.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. 2022.

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv: Computer Vision and Pattern Recognition*, 2021.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Animatable neural radiance fields for human body modeling. 2021a.

Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9054–9063, 2021b.

Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. *international conference on computer vision*, 2021.

Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 2017.

Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. Babel: Bodies, action and behavior with english labels. *computer vision and pattern recognition*, 2021.

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. *international conference on machine learning*, 2021a.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021b.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *international conference on machine learning*, 2016.

Aditya Sanghi, Hang Chu, Joseph G. Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *arXiv: Computer Vision and Pattern Recognition*, 2021.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pp. 894–906. PMLR, 2022.

Tian-Xiang Sun, Xiang-Yang Liu, Xi-Peng Qiu, and Xuan-Jing Huang. Paradigm shift in natural language processing. *Machine Intelligence Research*, 19(3):169–183, 2022.

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.

Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. *computer vision and pattern recognition*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. 2022.

Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate {3D} human pose in the wild using {IMUs} and a moving camera. *european conference on computer vision*, 2018.

Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Lafite: Towards language-free training for text-to-image generation. 2022.

A Appendix

A.1 Discussion and Limitations

As foundation models become more mature and learn more real-world knowledge, it provides us with new opportunities and challenges to a new learning paradigm. In this paper, we show one of the possibilities that learning from the foundation model instead of learning from data. We believe such attempts have an advantage over learning from data since the foundation model can better associate multi-modality data to make better decisions. Particularly, in our method, we found that using noisy training data can probe diverse knowledge out of the foundation model, which implies the feasibility of building an agent that can actively and continuously learn knowledge from the foundation model starting from chaos, i.e., noises, without manually feeding data which might limit the learnable knowledge of the foundation model. By this means, the agent might be able to learn something that is existed but we have not thought of yet or tasks we cannot formulate mathematically using our current knowledge. The reason why we investigate leveraging the foundation model to zero-shot motion generation is that we consider the foundation model as a promising world model of the real world, and the motion generation is one of the preliminary tasks for the agent to learn to interact with the real world.

However, as one of the few pioneers, there are several aspects that can be improved in our work. One is that CLIP learns from static image data and inherently lacks the capability to handle motion description. Using model dynamics learned from motion data can only compensate to a limited extent. It cannot handle some difficult texts like a sentence having multiple successive motions. Although this can be addressed by a divide-and-conquer strategy, it is a band-aid solution and resolves the problems once and for all. We suppose the best practice is to use a better foundation model that is able to handle the temporal description. There are several foundation models for aligning video and texts, but we found that most of them are learning with limited types of video data and are not as general as CLIP due to the difficulty of data collection for video training data. To this end, in our paper, we still prefer CLIP for zero-shot learning. And we leave the research with other foundation models in the future.

A.2 Visual Results

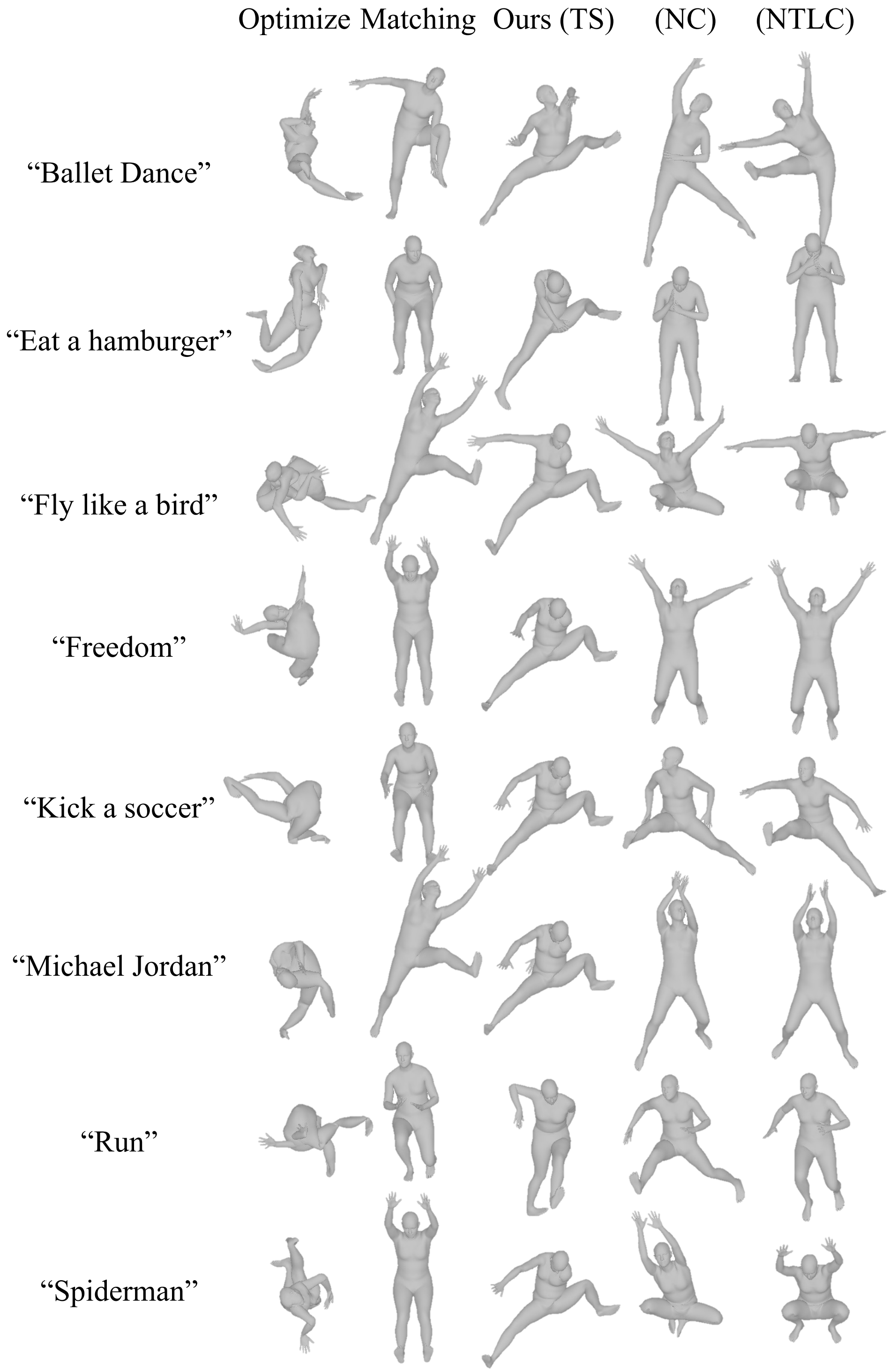

Figure 6: Visual results of text2pose baselines.

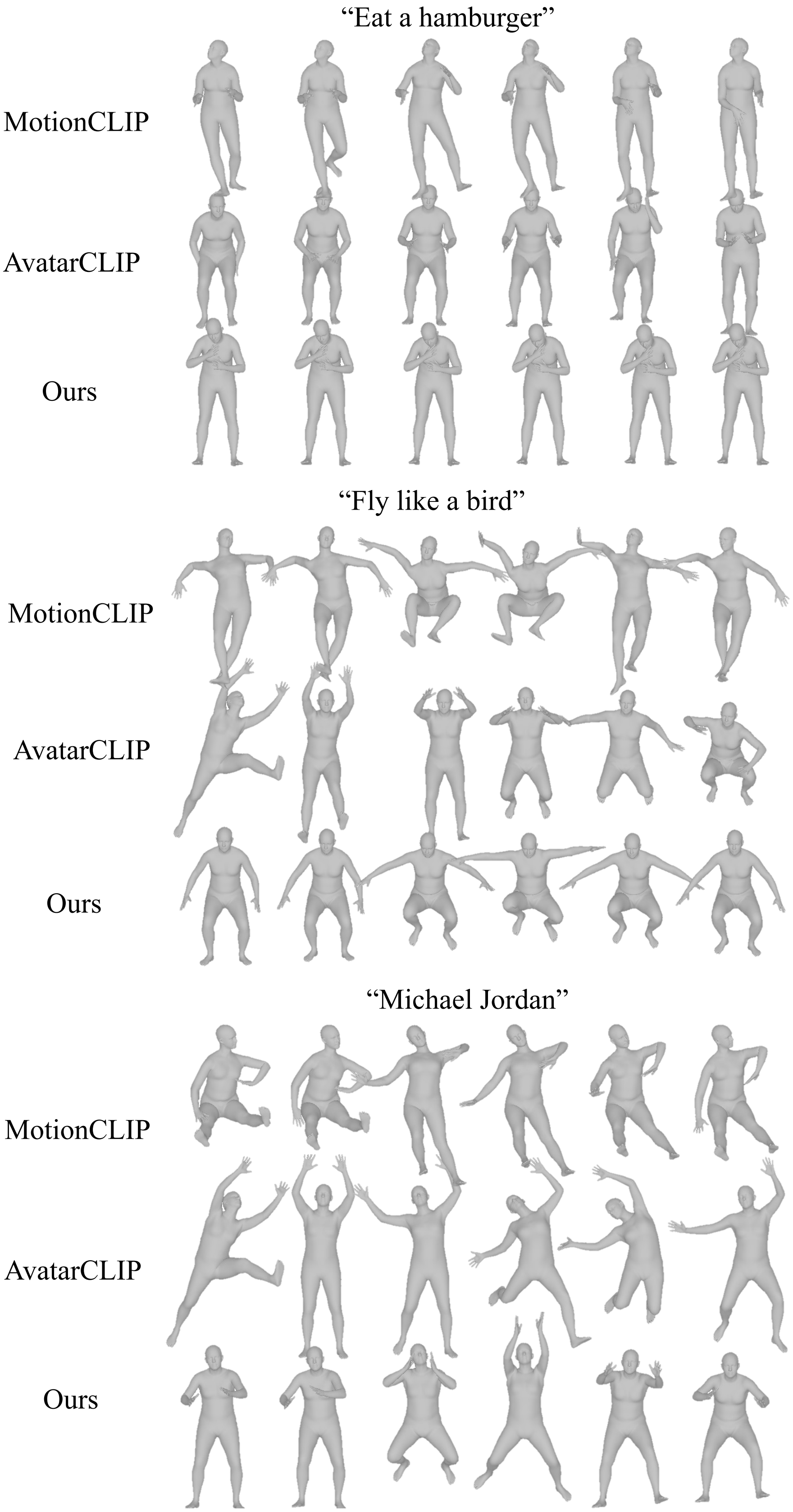

Figure 7: Visual results of text2motion baselines.