Being Comes from Not-being: Open-vocabulary Text-to-Motion Generation with Wordless Training

Junfan Lin[1,2] Jianlong Chang[3] Lingbo Liu[2] Guanbin Li[1] Liang Lin[1] Qi Tian[3*] Chang Wen Chen[2]
[1]Sun Yat-sen University [2]The Hong Kong Polytechnic University [3]Huawei Cloud

Abstract

Text-to-motion generation is an emerging and challenging problem, which aims to synthesize motion with the same semantics as the input text. However, due to the lack of diverse labeled training data, most approaches either limit to specific types of text annotations or require online optimizations to cater to the texts during inference at the cost of efficiency and stability. In this paper, we investigate offline open-vocabulary text-to-motion generation in a zero-shot learning manner that neither requires paired training data nor extra online optimization to adapt for unseen texts. Inspired by the prompt learning in NLP, we pretrain a motion generator that learns to reconstruct the full motion from the masked motion. During inference, instead of changing the motion generator, our method reformulates the input text into a masked motion as the prompt for the motion generator to "reconstruct" the motion. In constructing the prompt, the unmasked poses of the prompt are synthesized by a text-to-pose generator. To supervise the optimization of the text-to-pose generator, we propose the first text-pose alignment model for measuring the alignment between texts and 3D poses. And to prevent the pose generator from overfitting to limited training texts, we further propose a novel wordless training mechanism that optimizes the text-to-pose generator without any training texts. The comprehensive experimental results show that our method obtains a significant improvement against the baseline methods. The code is available at https://github.com/junfanlin/oohmg.

1. Introduction

Motion generation has attracted increasing attention due to its practical value in the fields of virtual reality, video games, and movies. Especially for text-conditional motion generation, it can largely improve the user experience if the virtual avatars can react to the communication texts in real time. However, most current text-to-motion approaches are trained on paired text-motion data with limited types of annotations, and thus could not well-generalize to unseen open-vocabulary texts.

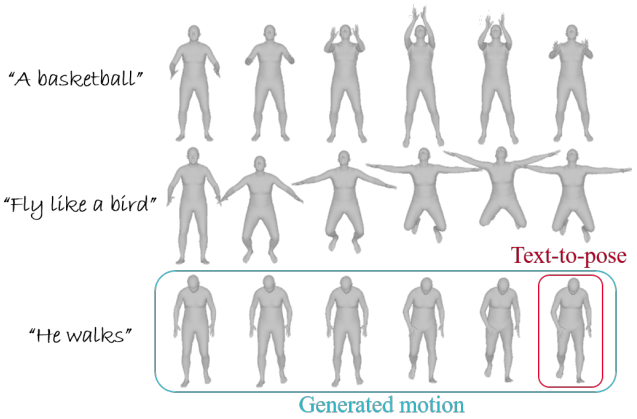

Figure 1. Demonstrations of our OOHMG. Given an unseen open-vocabulary text (e.g., an object name "a basketball", or a simile description "fly like a bird", or a usual text "he walks"), OOHMG translates the text into the text-consistent pose, which is used to prompt the motion generator for synthesizing the motion.

To handle the open-vocabulary texts, recent works leverage the powerful zero-shot text-image alignment ability of the pretrained model, i.e., CLIP [35], to facilitate the text-to-motion generation. Some works like MotionCLIP [42] use the CLIP text encoder to extract text features and learn a motion decoder to decode the features into motions. However, they require paired text-motion training data and still could not handle texts that are dissimilar to the training texts. Instead of learning an offline motion generator with paired data, some works like AvatarCLIP [13] generate motions for the given textual descriptions via online matching and optimization. Nevertheless, matching cannot generate new poses to fit diverse texts and online optimization is usually time-consuming and unstable.

In this paper, we investigate filling the blank of offline open-vocabulary text-to-motion generation in a zero-shot learning manner. For convenience, we term our method as **OOHMG** which stands for **O**ffline **O**pen-vocabulary **H**uman **M**otion **G**eneration. The main philosophy of

*Corresponding author: tian.qi1@huawei.com

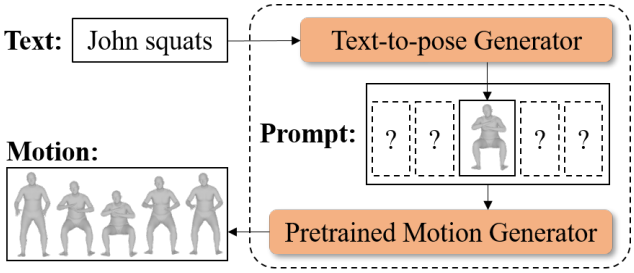

Figure 2. The sketch of OOHMG. A text is fed to the text-to-pose generator to obtain a text-consistent pose. Then, the pose is used to construct the motion prompt for the pretrained motion model to generate a motion.

OOHMG is inspired by prompt learning [5,19,41,48,50,52] in the field of natural language processing (NLP). Specifically, instead of changing the pretrained motion generator to cater to the given texts online, OOHMG reformulates the texts into a familiar input format to prompt the pretrained motion generator for synthesizing motions in the manner of "reconstruction". As for prompt construction, OOHMG learns a text-to-pose generator using the novel wordless training mechanism so that the pose generator can generalize to unseen texts during inference. After training, OOHMG uses the text-to-pose generator to translate texts into poses to construct the prompt. The overall sketch and demonstrations of OOHMG are illustrated in Fig. 2 and Fig. 1, respectively. In this sense, the two key ingredients of OOHMG include the motion generator pretraining and the prompt construction for open-vocabulary texts. In the following, we further elaborate on each of these ingredients.

As for the motion generator, we learn a motion generator by mask-reconstruction self-supervised learning. Particularly, our method adopts a bidirectional transformer-based [44] architecture for the motion generator. During training, the motion generator takes the randomly-masked motions as inputs and is optimized to reconstruct the original motions. To predict and reconstruct the masked poses from the unmasked, the motion generator is required to focus on learning motion dynamics which is the general need for diverse motion generation tasks. By this means, unlike previous methods that design different models for different tasks [1,2,11,18], our motion model can be directly applied to diverse downstream tasks by unifying the input of these tasks into masked motions to prompt the generator for motion generation. Moreover, our generator can flexibly control the generated content, such as the number, the order, and the positions of different poses of the generated motion by editing the masked motions, resulting in a controllable and flexible motion generation.

In constructing the motion prompt for open-vocabulary motion generation, OOHMG learns a text-to-pose generator and uses it to generate the unmasked poses of the masked motions, as shown in Fig. 2. There are two major difficulties in learning the text-to-pose generator: 1) what can associate diverse texts and poses to supervise the pose generator, and 2) how to obtain diverse texts as the training inputs. For difficulty 1, we build the first large-scale text-pose alignment model based on CLIP, namely TPA, that can efficiently measure the alignment between texts and 3D SMPL poses [27, 31] in the feature space. With TPA, the text-to-pose generator learns to generate poses for texts by maximizing the text-pose alignments via gradient descent. As for difficulty 2, instead of collecting massive texts laboriously for training, we consider an extreme training paradigm, termed wordless training. Just as its name implies, wordless training only samples random training inputs from the latent space of texts. And we found that the optimized pose generator can well-generalize to real-world texts.

Overall, the contributions of OOHMG are as follows. 1) We propose an offline open-vocabulary text-to-motion generation framework, inspired by prompt learning, and 2) to supervise the training process of the text-to-pose generator, we propose the first text-pose alignment model, i.e., TPA, and 3) to endow the text-to-pose generator with the ability to handle open-vocabulary texts, we train the generator with the novel wordless training mechanism. 4) Extensive experiment results show that OOHMG is able to generate motions for open-vocabulary texts efficiently and effectively, and obtain clear improvement over the advanced baseline methods qualitatively and quantitatively.

2. Related Work

Conditional Motion Generation can be classified into various categories based on the types of conditions. For example, music has been utilized as a condition in some studies to generate dance motions [1], while others have synthesized movements through short motion descriptions [2, 3, 18] and action labels [11,30]. The success of these methods is heavily dependent on large motion capture datasets [6, 7, 15, 21, 22, 43, 47] and labeled motion description datasets, including AMASS [34], KIT motion-language dataset [32], and HumanML3D dataset [10]. However, such datasets are often limited by their task design and data collection challenges, such as the failure to account for emotional movements. Although several methods have demonstrated impressive qualitative and quantitative results [4, 53], those trained on limited datasets are unable to generalize to open-vocabulary motion descriptions.

Probing Knowledge from Pretrained Model. The development of pretrained foundation models has led to the potential for zero-shot/few-shot learning to surpass supervised learning [5, 8, 36, 49]. One such model, CLIP [36], has the ability to semantically align language-vision latent spaces [46]. Combined with CLIP, DALL-E [35] enables impressive text-to-image synthesis capabilities. This powerful representation ability of foundation model has led to

the emergence of zero-shot text-driven applications [9, 14, 26, 29], including 3D meshes generation [16, 17, 23, 28, 39].

Related to ours, recent studies [37, 38] have combined CLIP with diffusion generation models to generate text-consistent 3D meshes [16, 33], while other methods focus on generating static meshes or 2D images for text-video generation, e.g. Make-A-Video [40], Imagen Video [12], and Phenaki [45]. As for open-vocabulary motion generation, CLIP-Actor [51] simply uses motion from existing datasets by matching the textual descriptions with the motion labels of the existing text-motion datasets. And MotionCLIP [42] learns a motion VAE by regularizing the latent space to align with the feature space of CLIP, which also requires labeled data. AvatarCLIP [13] is the closest method to ours, as it also explores the potential of zero-shot open-vocabulary motion generation, but our approach does not require online matching or optimization.

3. Preliminaries

This paper investigates offline zero-shot open-vocabulary human motion generation (OOHMG). To address open-vocabulary texts, OOHMG includes a text-pose alignment model based on the text-image alignment model, i.e., CLIP [35]. In this section, we provide a brief introduction to the task as well as CLIP.

Open-vocabulary 3D human motion generation involves generating a motion m that aligns with a given natural language motion description d, such as "fly like a bird." A motion is a sequence of 3D poses, $m = [p_t]_{t=1:T}$, where p is the 3D pose, t represents the timestep, and T is the maximum length of the motion. We use a 6D-rotation representation $p \in \mathbb{R}^{J\times 6}$ [54], but we also utilize the latent representation $p^l \in \mathbb{R}^{32}$ of VPoser [27] which is a well-known pose VAE trained on massive poses, to incorporate the pose prior from VPoser. Our focus is on generating body motion, and therefore we do not consider facial expressions, hand poses, or global orientation. We utilize SMPL [20], a popular parametric human model, for its interpretability and compatibility with various platforms. SMPL is a parametric human model driven by large-scale aligned human surface scans [31], and feeding pose representations into SMPL enables us to obtain 3D meshes $v = \mathcal{M}_{\text{SMPL}}(p)$.

CLIP [35] is a vision-language pre-trained model designed for large-scale image-text datasets. It comprises two encoders: an image encoder E_o and a text encoder E_d. We use o to denote the image and d to represent the text. The encoders are trained such that the latent codes of paired images and texts are pulled together, while the unpaired codes are pushed apart. Formally, the CLIP loss function is

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

where $\mathbf{o}$ and $\mathbf{d}$ is the sets of images $\{o_i\}_{i=1:B}$ and texts

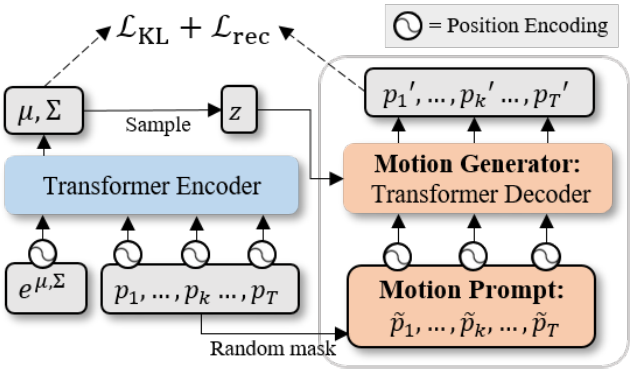

Figure 3. Training process of motion generator. The motion encoder takes in the original motion and extracts the latent feature. The motion generator takes in masked motion and the latent feature to reconstruct the original motion.

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

$$\mathbf{Pr}(o_i|d_i) = \frac{\exp\Big(\text{cossim}\big(E_o(o_i), E_d(d_i)\big)/H\Big)}{\sum_{j=1:B} \exp\Big(\text{cossim}\big(E_o(o_j), E_d(d_i)\big)/H\Big)}, \quad (2)$$

where $\text{cossim}(f_i, f_j) = \frac{f_i^T f_j}{|f_i||f_j|}$, and H is the temperature to adjust the sensitivity of softmax. The calculation of $\mathbf{Pr}(d_i|o_i)$ follows the similar process. For convenience, we use **CLIP score** to stand for the cosine similarity between text and image features from CLIP.

4. OOHMG

As mentioned above, our OOHMG achieves offline open-vocabulary text-to-motion generation with two key ingredients, i.e., the pretrained motion generator and prompt construction. Both of these components manage to be text-free during the training phase. In this section, we detail these two modules formally.

4.1. Motion Generator Pretraining

In advanced language modeling in NLP, the language model [8] learns to reconstruct the masked sentence from the randomly masked sentence in a self-supervised manner. Our motion generator also follows a similar training strategy. Specifically, during training, the random proportion of the poses of a motion $m = [p_t]_{t=1:T}$ are masked by a learnable embedding $e^{\text{mask}} \in \mathcal{R}^{|p|}$. Formally, the pose $\tilde{p}_t$ of the masked motion $\tilde{m}$ 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 equals 1, the original pose is preserved. Otherwise, the pose is replaced by the mask embedding. In addition, since there is usually more than one motion corresponding to the same masked motion, to prevent learning a generator that generates the average motions, we also adopt a motion encoder to extract the latent feature for each motion. The motion encoder follows ACTOR [30] without the motion category conditions, as illustrated in Fig. 3. The motion generator takes in $\tilde{m}$ and the latent code from the motion encoder to predict the m' to reconstruct the original m. Different from ACTOR which is optimized to encode and decode the full sequence, ours is to predict the full sequence from the masked sequence. Hence, to meet different requirements, an ACTOR-based generator, e.g., AvatarCLIP, needs to search in the latent space, which might be inefficient and unstable. Instead, our method can control the generation via motion prompt explicitly, which is more transparent and controllable. During inference, the motion encoder is discarded and the latent feature can be randomly sampled from $\mathcal{N}(0; 1)$. Formally, the loss function $\mathcal{L}_{m2m}$ for the motion generator is

$$\begin{aligned} \mathcal{L}_{rec}(p'_t, p_t) &= ||p_t - p'_t||_2 + ||v_t - v'_t||_2 \\ \mathcal{L}_{m2m}(m', m) &= \sum_{p'_t, p_t \in m', m} \mathcal{L}_{rec}(p'_t,\ p_t) + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}, \end{aligned} \quad (3)$$

where $v = \mathcal{M}_{\text{SMPL}}(p)$ and $\mathcal{L}_{\text{KL}}$ is the KL-divergence regularization term to pull the predicted latent features to the normal distribution. After pretraining, the motion generator can be used to generate motion for downstream tasks by using the motion prompts, i.e., masked motions.

4.2. Prompt Construction

Since the mask in the motion prompt is provided by the motion generator, we only need to synthesize the unmasked poses to construct the motion prompt. In other words, we should use texts to synthesize the unmasked poses. To achieve this, our OOHMG learns a text-to-pose generator that takes in texts and predicts the poses. To provide supervision during training, we propose the first text-pose alignment model, TPA, based on the large-scale text-image alignment model CLIP. And to cover as diverse text as possible, we adopt an extreme yet effective training paradigm, i.e., wordless training. Below, we detail each procedure.

Text-pose alignment model. Due to the lack of massive paired text-pose data, it is non-trivial to learn TPA from scratch. To this end, our TPA reuses the text encoder of CLIP. As for the pose encoder, TPA mines 3D pose knowledge of the CLIP image encoder. In fact, TPA is not the first work to leverage the CLIP image encoder for pose feature extraction. In AvatarCLIP [13], the researchers extract pose features via the pipeline "pose→generate SMPL meshes→render multi-view images→use CLIP to extract image features→average the features to obtain the final image features", as shown in the upper part of Fig. 4. As testified in AvatarCLIP, this strategy should be enough for zero-shot text-pose alignment. And in our experiments, we also find it enough for our generator to learn to generate visually-plausible and text-consistent poses. Nevertheless, we do think it can be further improved by preserving the angle information using techniques like the view-dependent conditioning of DreamFusion [33] Unfortunately, AvatarCLIP [13] found this pipeline difficult to supervise the pose generation. In [16], the researchers found that, if the generation space is too unconstrained, training solely with CLIP loss will result in severe artifacts that satisfy CLIP loss but with unrealistic geometry like Deep Dream artifacts [24]. The potential reason is that CLIP has been trained on diverse images and there might be diverse solutions for the same text, causing the optimization divergent. To address this problem, our TPA limits the solution space to the 3D poses by distilling a tailored yet specific pose encoder. Specifically, we adopt an end-to-end pose encoder E_p for mapping the poses p to their features f_p^* as shown in Fig. 4. The distillation objective $\mathcal{L}_{E_p}$ is:

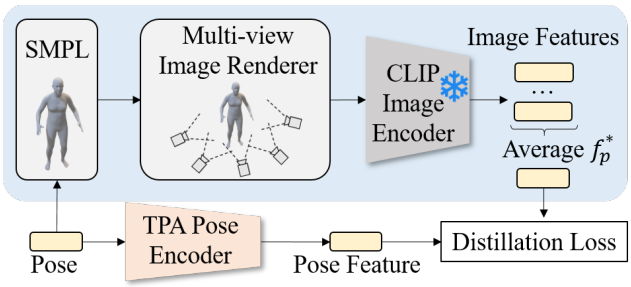

Figure 4. The learning process of the TPA pose encoder. The upper part is the pipeline to extract the pose features via reusing the CLIP image encoder. TPA learns an end-to-end pose encoder that takes in the poses and predicts the output of the pipeline.

$$\mathcal{L}_{E_p}(p, f_p^*) = ||E_p(p) - f_p^*||_2 - \text{cossim}\big(E_p(p), f_p^*\big), \quad (4)$$

where the first term of Equ.(4) is for reducing the element-level distance between features. While the second term of Equ.(4) is to reduce the angular difference.

Wordless training for generalized text-to-pose generator. To generate poses for open-vocabulary texts, the text-to-pose generator should train with as diverse texts as possible. However, as the text space is combinatorial, it's impractical to enumerate all possible texts for training. Nevertheless, since the texts will be encoded into text features by the text encoder of TPA to measure the alignment with the pose, it occurs to us that we can directly build a text-to-pose generator upon the normalized text feature space of TPA instead of real-world text space. By this means, it becomes trivial to obtain diverse inputs for training. To obtain

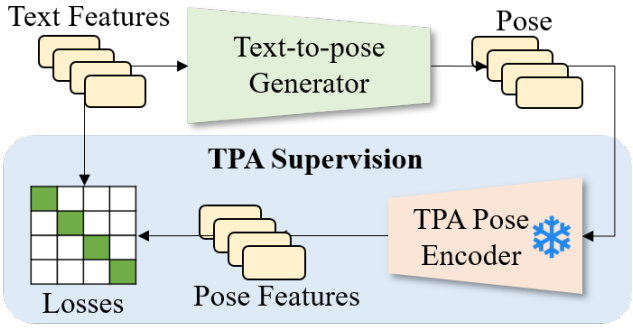

Figure 5. Training process of text-to-pose generator. During training, the input features are randomly sampled from the feature space of the text encoder.

diverse text features f_d, we sample them from Normal or Uniform distribution randomly by:

$$f_d = \frac{\epsilon + b}{|\epsilon + b|}, \quad \forall \epsilon_i \in \epsilon, \epsilon_i \sim \mathcal{N}(0;1) \text{ or } \mathcal{U}[-1,1], \tag{5}$$

where $b \sim \mathcal{U}[-1,1]$ is a random bias to avoid the features sampled around the zero features. We don't take the scale into account since there is a normalization operator.

Our text-to-pose generator G_{t2p} takes a batch of sampled features $\mathbf{f_d} = [f_{d_1}, ..., f_{d_B}]$ as the inputs and predict the latent poses $\mathbf{p}^l = G_{\text{t2p}}(\mathbf{f_d})$ of VPoser. As mentioned in the preliminary, $\mathbf{p}^l$ can be decoded into the poses $\mathbf{p}$ by the decoder of VPoser. We can regulate the generator to predict in-distribution poses of VPoser by pulling the $\mathbf{p}^l$ close to the prior distribution of VPoser, i.e., the normal distribution. The optimization target is to minimize:

$$\mathcal{L}_{\text{t2p}}(\mathbf{p}^l, \mathbf{p}, \mathbf{f_d}) = \mathcal{L}_{\text{TPA}}\Big(E_p(\mathbf{p}), \mathbf{f_d}\Big) + \lambda_{\text{L2}}||\mathbf{p}^l||_2, \tag{6}$$

where the second term of the loss function is used to regulate the predicted latent pose to close to the prior distribution of VPoser. And $\mathcal{L}_{\text{TPA}}$ is the same as $\mathcal{L}_{\text{CLIP}}$ in Equ.(1) with image features replaced by pose features. The optimization process is illustrated in Fig. 5.

Overall Training Procedure. Different modules of our method are trained separately since they have different training data. Specifically, we train the motion generator and TPA using AMASS data. After that, we train the text-to-pose generator with the frozen TPA and the wordless training strategy. During inference, a text is first encoded by the CLIP text encoder into text features. Then, the text-to-pose generator generates the text-consistent pose according to the text features, which are used for constructing the motion prompt. And the motion prompt will drive the motion generator to reconstruct the full motion.

5. Experiments

We first introduce the datasets and baseline methods used in our experiments. Next, we evaluate the overall performance of zero-shot open-vocabulary human motion generation. And we also compare the performance of text-to-pose generation. After that, we conduct ablation studies to better understand our method.

General Settings. In our experiments, all motion data and textual descriptions originated from AMASS [21] and BABEL [34], respectively. AMASS unifies various optical marker-based mocap datasets with more than 40 hours of motion data without textual labels. Following the same settings in [13,30], we down-sample the motion capture framerate to 30 per second and limit the duration of a motion to 2 seconds. As for the text data, BABEL is a dataset of textual sentences for motions. We remove lengthy sentences which exceed the CLIP's maximum text length of 77, resulting in a dataset with a size of 4178. In our paper, the checkpoint of CLIP ("CLIP-ViT-B/32") is used. More about training details such as hyperparameters are presented in the supplementary. The code will be released.

Baselines. In the following, we enumerate the related baseline methods. To the best of our knowledge, our work is the first offline zero-shot open-vocabulary text-to-motion generation. Therefore, we include a baseline of online zero-shot open-vocabulary text-to-motion generation, i.e., AvatarCLIP [13], and a baseline of offline supervised open-vocabulary text-to-motion generation, i.e., MotionCLIP [42]. Similar to ours, AvatarCLIP also includes a text-to-pose phase via matching, and uses the matched poses to search the motion in the latent space of a pretrained motion VAE. Therefore, in evaluating text-to-pose generation, matching, as well as other baselines considered in AvatarCLIP [13], are also included in our experiments. As for our OOHMG, except for the experiment for measuring controllability, we generate poses for texts and place the generated poses in the middle of the masked motions. More details about the baselines are placed in the supplementary.

5.1. Open-vocabulary Text-to-Motion Generation

In this part, we are interested in testifying about the ability to generate text-consistent motion across different baselines. For this purpose, we evaluate 1) whether the generated contents follow real-world motion dynamics, and 2) whether the generated motions are text-consistent.

Motion Dynamics. To answer the first question, we propose to measure the distance between the generated contents and the distribution of the real-world motion. Therefore, we train a general motion VAE upon AMASS, and use the average reconstruction error of the motion VAE to indicate the in-distribution degree (In-distrib.) of the baselines. Please refer to the supplementary for more details about the motion VAE. The larger the In-distrib. is, the more distant the generated contents are to the real-world motion distribution. We use BABEL as textual descriptions to generate motions. As reported in Tab. 1, our method outperforms

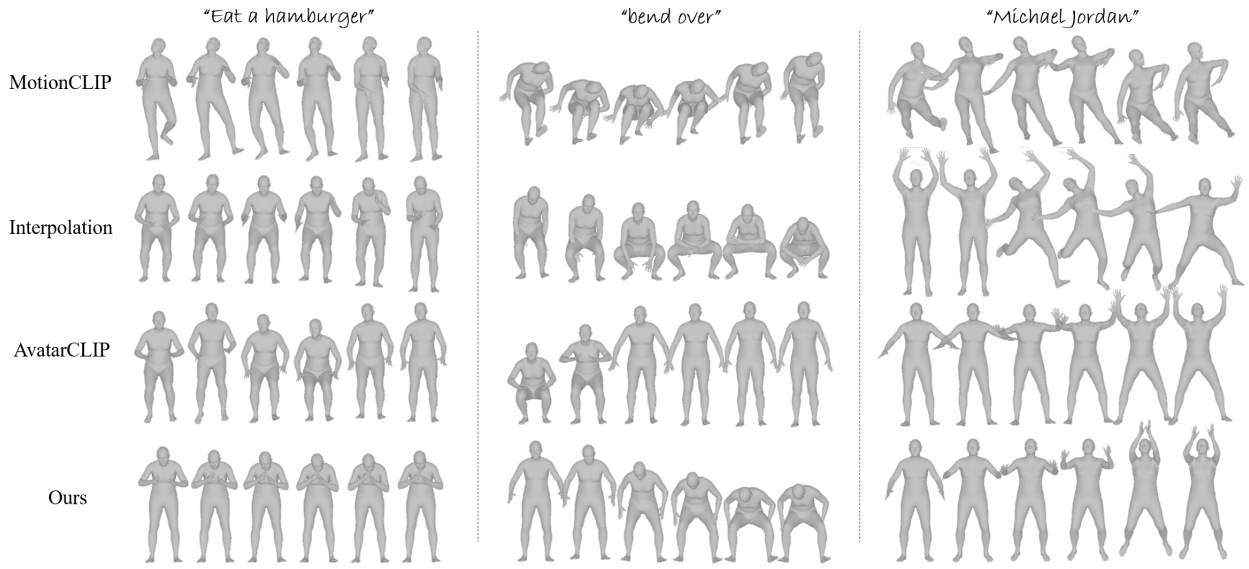

Figure 6. The visual comparison results of different open-vocabulary text-to-motion methods. The results are part of the generated motions due to the space limit.

Table 1. Comprehensive results for open-vocabulary text-to-motion generation. The arrow ↑ indicates the performance is better if the value is higher.

	In-distrib.↓	Top1↑	Top10↑	Top50↑
MotionCLIP [42]	0.2191	0.0029	0.0153	0.0661
Interpolation [13]	0.0312	0.0045	0.0472	0.1927
AvatarCLIP [13]	0.0407	0.0002	0.0069	0.0290
Ours	**0.0205**	**0.0792**	**0.3231**	**0.6494**

the others by a clear margin. Particularly, we found that Interpolation [13] generates motion through linear interpolation without considering the motion dynamics, resulting in poor In-distrib. results. We also notice that the parameterized MotionCLIP performs poorly. The potential reason might be the gap of latent space between MotionCLIP and the CLIP text encoder. As shown in Fig. 6, we observe that MotionCLIP is likely to generate twisted poses while AvatarCLIP and ours are more natural.

Generating Text-consistent Motion. Since both related baselines MotionCLIP and AvatarCLIP are CLIP-based, we use CLIP to measure the alignment between the motions and the texts. To this end, we extend the CLIP-R-precision [25] to the level of motion to measure the text-motion alignment. Specifically, we say that the text-motion matching is accurate if, among all poses of the generated motion from different texts, the best-matched pose of the text is located in the generated motion of the text. To achieve a better motion-level CLIP-R-precision, the generated motion should **1) contain the text-consistent poses, 2) and does not contain irrelevant poses that might cause mismatching for other poses**. From the results at the right of Tab. 1, we find that among all baselines, our method obtains the best TopK motion-level CLIP-R-Precision by a clear margin. It is worth noting that our method does not require online matching or optimization or paired text-motion training data like the baseline methods. We also observe that Interpolation [13] performs better than AvatarCLIP. One of the reasons is that, unlike Interpolation [13] which includes the condition poses as part of the generated motion, AvatarCLIP requires online optimization to obtain the motion, which is unstable and non-trivial to generate a motion consisting of the condition poses. And Interpolation [13] is less likely to generate new poses that might distract the matching process.

5.2. Prompt construction

As described above, the prompt is in the form of masked motion. And OOHMG uses a text-to-pose generator to synthesize the unmasked poses of the masked motion according to the texts. Therefore, we are interested in 1) whether the generated motion can be controlled by the motion prompt, and 2) whether the generated poses are text-consistent.

Controllability. To answer question 1, we use the 4096 clustered poses used in AvatarCLIP [13] as the condition poses. We randomly sample from the clustered poses to construct the *K*P test set, where $K \in \{1, 2, 3\}$ indicates the number of the unmasked poses of the motion prompt. We calculate the distance between *K* poses and the closest poses of the generated motions. As shown on the left of Tab. 3, we observe that our method also possesses the best controllability. Notice that, Interpolation [13] directly takes the condition poses as a part of the generated motion. However, there is still a small error for Interpolation [13] since interpolation is conducted on the latent codes of these poses in the latent

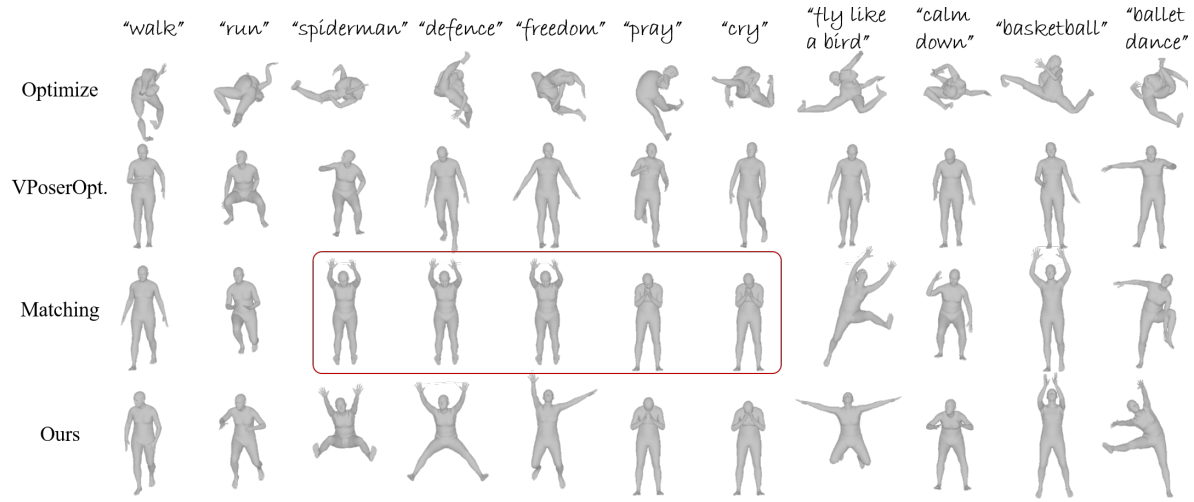

Figure 7. The visual results of different text-to-pose generation methods. Besides ours, the other baseline methods require online matching or optimizations.

Table 2. Comparison among text-to-pose baselines. The arrow ↑ indicates the performance is better if the value is larger.

	CLIP Score ↑	In-distrib. ↓	Cyc. Loss ↓	Top1 ↑	Top10 ↑	Top50 ↑
Matching [13]	0.2615	**0.0015**	0.0288	0.0127	0.0831	0.2820
Optimize [13]	0.2455	0.8365	0.0047	0.0005	0.0038	0.0120
VPoserOptimize [13]	0.2460	**0.0015**	0.0048	0.0005	0.0029	0.0168
Ours	**0.2694**	**0.0015**	**0.0045**	**0.0775**	**0.3284**	**0.6711**

Table 3. Controllability for pose-conditioned motion generation.

	1p↓	2p↓	3p↓
Interpolation [13]	0.0900	0.0865	0.0868
AvatarCLIP [13]	0.8323	1.5982	2.1447
Ours	**0.0452**	**0.0131**	**0.0129**

Table 4. Ablation results for our text-to-pose generator.

	CLIP Score ↑	In-distrib. ↓	Top50 ↑
VPoserOptimize [13]	0.2460	**0.0015**	0.0168
Ours (Text+Score)	0.2620	0.1127	0.2090
Ours (Text+$\mathcal{L}_{\text{TPA}}$)	0.2601	0.1210	0.2104
Ours (Random+$\mathcal{L}_{\text{TPA}}$)	**0.2711**	0.0111	**0.7224**
Ours (Random+$\mathcal{L}_{\text{t2p}}$)	0.2694	**0.0015**	0.6711

space of VPoser and the encode-decode process causes the error. Differently, our method directly generates the motion and manages to obtain a smaller *K*P error. Nevertheless, we found that AvatarCLIP is difficult to generate motion that well-preserves the given poses. The potential reason is that AvatarCLIP requires optimizing the motion latent code in the high-dimensional latent space, which might be nonconvex and require a large number of optimization steps.

Open-vocabulary Text-to-Pose Generation. To comprehensively understand our text-to-pose generator, we evaluate the generated poses from four aspects, i.e., the text-pose alignment (CLIP Score), the distance to the real-world pose distribution (In-distrib.), how much text information is preserved in the generated poses (Cycle loss for reconstructing text features from the generated poses) and CLIP-R-precision [25] (TopK). For a detailed explanation of different metrics, please refer to the supplementary. The baseline methods are adopted from AvatarCLIP since it is the only work that includes the zero-shot open-vocabulary text-to-pose generation, to our best knowledge. The results are represented in Tab. 2. As we can see, the Matching method can obtain a higher CLIP score than Optimize and VPoserOptimize, which implies directly using CLIP to match is more effective than optimization via the complex pipeline as depicted in Fig 4, which is also mentioned in AvatarCLIP. Nevertheless, Matching is unable to generate more accurate poses for diverse texts and therefore is less capable of preserving text information in the generated poses (i.e., high Cyc. loss in Tab. 2). As shown in the red-circled region in Fig. 7, Matching uses the same poses for texts with different meanings. By using TPA and wordless training, our text-to-pose generator obtains significant improvement across various metrics. Notice that, different from the baseline methods, our method does not require any online matching/optimization and does not see any real-world texts during the training phase. It means that our generator can be well-generalized to real-world texts.

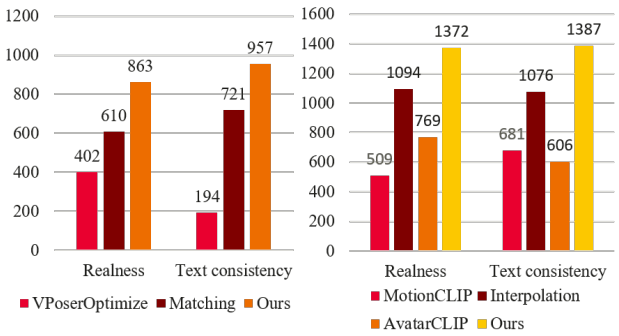

Figure 8. Human evaluation for text-to-pose generation (Left) and text-to-motion generation (Right).

Ablation studies. As we can see in Tab. 4, by simply replacing the original pipeline with TPA, the text-to-pose generator that optimizes to maximize the TPA score, i.e., Ours (Text+Score), can significantly improve the CLIP score in comparison with VPoserOptimize. However, during our experiment, we observed that maximizing the TPA score as the objective is sensitive to the performance of TPA. And the stability of optimization can be further improved when minimizing the $\mathcal{L}_{\text{TPA}}$ instead (analysis in the supplementary). We contribute such stability to more dense supervision by drawing other samples into the contrastive loss. To this end, we suggest using $\mathcal{L}_{\text{TPA}}$ for the other experiments. Another valuable observation is that, by wordless training with randomly sampled text features, Ours (Random+$\mathcal{L}_{\text{TPA}}$) not only achieves the best CLIP score but also obtains significant improvement on the In-distrib. metric. The potential reason could be the infinite amount and diversity of training data. With a limited amount of training data, it's easier for the generator to exploit the difference between TPA and the original pipeline to obtain an overfit solution (e.g., generating strange or twisted poses) for the training data. Another evidence is that, by using wordless training, Ours (Random+$\mathcal{L}_{\text{TPA}}$) has better performance than Ours (Text+$\mathcal{L}_{\text{TPA}}$). It implies that using real texts for training might encourage the generator to overfit TPA, resulting in a poor CLIP score. To generate in-distribution poses, Ours (Random+$\mathcal{L}_{\text{t2p}}$) also includes the L2-norm regularization term which pulls the predicted latent pose to the center of the prior distribution of VPoser.

5.3. Qualitative Results

Human Evaluation. As for the qualitative results, We also conducted a series of human evaluations. We design a questionnaire that includes 50 queries for comparing different methods (25 for text-to-pose generation and 25 for text-to-motion generation). In each query, the participant was required to rank the performance of different methods in terms of realness and text consistency. For pose generation, we assign scores 2, 1, and 0 for the methods with ranks 1st, 2nd, and 3rd, respectively. And for motion generation, we assign scores 3, 2, 1, and 0 for the methods ranking 1st, 2nd, 3rd, and 4th, respectively. By the end of the submission, we have collected 25 available feedbacks and the total scores for each method are calculated and reported in Fig. 8. For more details, please refer to the supplementary. From the results, we observe that our methods for both pose and motion generations have obtained the best results. And we also find that these results are mostly in line with the quantitative results in previous experiments. It suggests that future works can follow the same evaluation protocol for this task.

Table 5. The inference efficiency of different methods.

	Batch size ↑	Time (sec) ↓
Pipeline with CLIP [13]	15	1.2068
Our TPA	∼ **130K**	**0.0172**
MotionCLIP [42]	375	0.0242
AvatarCLIP [13]	9	140
Our OOHMG	∼**14K**	**0.0159**

Efficiency. Here, we also compare the maximum samples each method can handle simultaneously (i.e., batch size) and the time cost. The reported time is per batch and the batch size is 1, and is averaged over 100 repeated experiments. The experiments are conducted using one NVIDIA V100 Tensor Core (32G). As shown in Tab. 5, both our TPA and OOHMG can directly infer the results with significant improvements. As the results suggest, our OOHMG is the first real-time zero-shot text-to-motion generation method.

6. Conclusion

In this paper, we propose an offline open-vocabulary human motion generation (OOHMG) framework in a zero-shot learning manner, which draws inspiration from prompt learning. To address the difficulty of optimization with the complex pipeline, we propose the first text-pose alignment model which is efficient and effective for supervising the training of the pose generator. To handle diverse and unseen real-world texts, we propose a novel wordless training mechanism. Extensive experiments show that our method can generate better text-consistent poses and motions across various baselines and metrics.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant No.2021ZD0111601, in part by the Guangdong Basic and Applied Basic Research Foundation (NO. 2020B1515020048), in part by the National Natural Science Foundation of China (NO. 61976250), in part by the Shenzhen Science and Technology Program (NO. JCYJ20220530141211024) and in part by the Fundamental Research Funds for the Central Universities under Grant 22lgqb25.

References

[1] Gunjan Aggarwal and Devi Parikh. Dance2music: Automatic dance-driven music generation. *arXiv: Sound*, 2021. 2

[2] 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. 2

[3] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. *international conference on 3d vision*, 2019. 2

[4] Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Güll Varol. TEACH: Temporal Action Compositions for 3D Humans. In *International Conference on 3D Vision (3DV)*, 2022. 2

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[6] 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. 2

[7] 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. 2

[8] 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. 2, 3

[9] 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. 3

[10] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022. 2

[11] 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. 2

[12] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 3

[13] 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*, 2022. 1, 3, 4, 5, 6, 7, 8

[14] Ricong Huang, Weizhi Zhong, and Guanbin Li. Audio-driven talking head generation with transformer and 3d morphable model. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 7035–7039, New York, NY, USA, 2022. Association for Computing Machinery. 3

[15] 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. 2

[16] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, pages 867–876, 2022. 3, 4

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

[18] Xiao Lin and Mohamed R Amer. Human motion modeling using dvgans. *arXiv preprint arXiv:1804.10652*, 2018. 2

[19] Lingbo Liu, Bruce XB Yu, Jianlong Chang, Qi Tian, and Chang-Wen Chen. Prompt-matched semantic segmentation. *arXiv preprint arXiv:2208.10159*, 2022. 2

[20] 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. 3

[21] 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. 2, 5

[22] 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. 2

[23] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. 2022. 3

[24] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017. 4

[25] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. *neural information processing systems*, 2021. 6, 7

[26] 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. 3

[27] 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. 2, 3

[28] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Animatable neural radiance fields for human body modeling. 2021. 3

[29] 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*, pages 9054–9063, 2021. 3

[30] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. *international conference on computer vision*, 2021. 2, 4, 5

[31] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 2017. 2, 3

[32] Matthias Plappert, Christian Mandery, and Tamim Asfour. The kit motion-language dataset. *Big Data*, 2016. 2

[33] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 3, 4

[34] 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. 2, 5

[35] 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*, pages 8748–8763. PMLR, 2021. 1, 2, 3

[36] 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*, 2021. 2

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj
”orn Ommer. High-resolution image synthesis with latent diffusion models. 2022. 3

[38] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar, Seyed Ghasemipour, Burcu Karagol, S Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. 2022. 3

[39] 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. 3

[40] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 3

[41] 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. 2

[42] 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, 3, 5, 6, 8

[43] 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. 2

[44] 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. 2

[45] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022. 3

[46] 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. 2

[47] 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. 2

[48] Haixin Wang, Jianlong Chang, Xiao Luo, Jinan Sun, Zhouchen Lin, and Qi Tian. Lion: Implicit vision prompt tuning, 2023. 2

[49] Likang Wang, Yue Gong, Xinjun Ma, Qirui Wang, Kaixuan Zhou, and Lei Chen. Is-mvsnet: Importance sampling-based mvsnet. In *European Conference on Computer Vision*, pages 668–683. Springer, 2022. 2

[50] Shijie Wang, Jianlong Chang, Zhihui Wang, Haojie Li, Wanli Ouyang, and Qi Tian. Fine-grained retrieval prompt tuning. *arXiv preprint arXiv:2207.14465*, 2022. 2

[51] Kim Youwang, Kim Ji-Yeon, and Tae-Hyun Oh. Clip-actor: Text-driven recommendation and stylization for animating human meshes. *arXiv preprint arXiv:2206.04382*, 2022. 3

[52] Bruce XB Yu, Jianlong Chang, Lingbo Liu, Qi Tian, and Chang Wen Chen. Towards a unified view on visual parameter-efficient transfer learning. *arXiv preprint arXiv:2210.00788*, 2022. 2

[53] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 2

[54] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019. 3