THU-PM-250

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

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

CVPR2023 Highlight

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

中山大學 SUN YAT-SEN UNIVERSITY

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

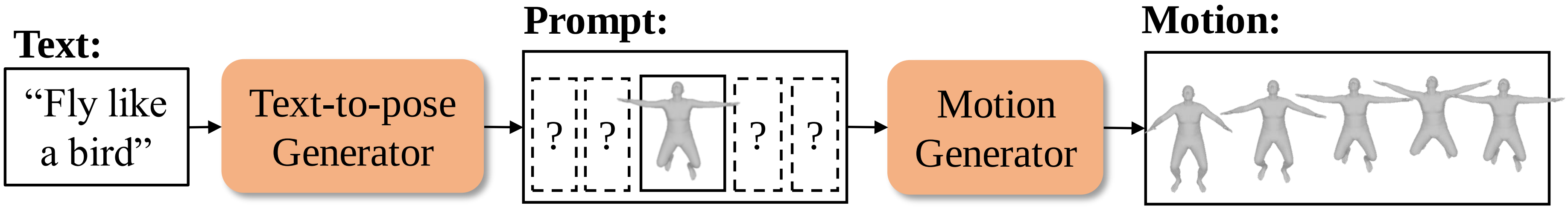
HUAWEI

OOHMG overview

- **OOHMG** stands for **O**ffline **O**pen-vocabulary **H**uman **M**otion **G**eneration:
  - **O**ffline: **Do not requires online finetuning/matching** to handle unseen texts **in real time**
  - **O**pen vocabulary: Generate poses/motions for **any type of texts** in a **zero-shot** manner
  - **H**uman **M**otion **G**eneration: Generate realistic motion with **SMPL (a 3D human model)**

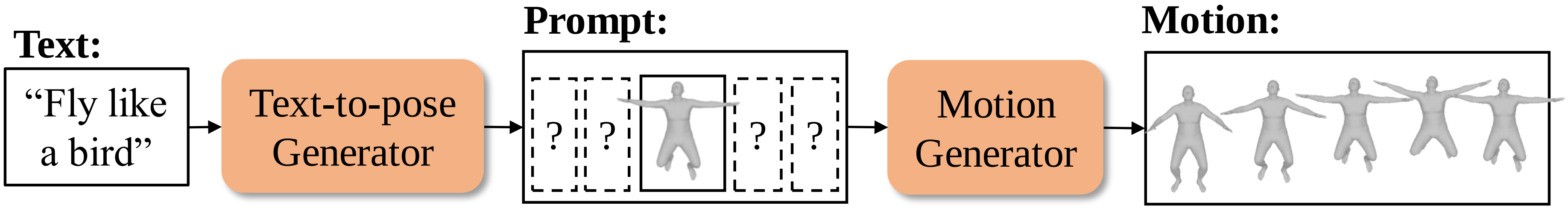
OOHMG overview

- OOHMG stands for Offline Open-vocabulary Human Motion Generation:
  - Offline: Do not requires online finetuning/matching to handle unseen texts in real time
  - Open vocabulary: Generate poses/motions for any type of texts in a zero-shot manner
  - Human Motion Generation: Generate realistic motion with SMPL (a 3D human model)



OOHMG overview

- OOHMG stands for Offline Open-vocabulary Human Motion Generation:
  - Offline: Do not requires online finetuning/matching to handle unseen texts in real time
  - Open vocabulary: Generate poses/motions for any type of texts in a zero-shot manner
  - Human Motion Generation: Generate realistic motion with SMPL (a 3D human model)

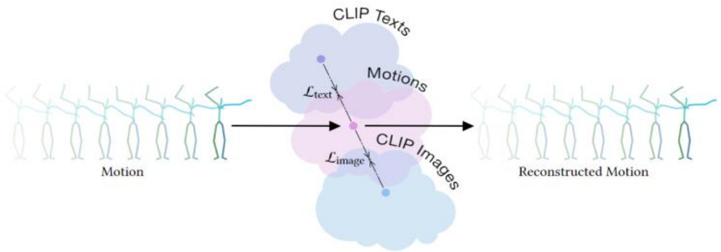


- The three key ingredients of OOHMG include:
  - **Text-pose alignment:** a differentiable and effective alignment model for 3D poses and texts, which is distilled from the pretrained language-Image alignment model, i.e., CLIP.
  - **Wordless training:** a training framework that optimizes the pose generator with random text features, which generalizes the generator to handle unseen real-world texts.
  - **Pose-prompt motion generator:** the motion generator learns to reconstruct a masked motion sequence. By this means, the motion generation can be controlled by reformulating the text-consistent poses as a masked motion to prompt the generator.

Methodology

Previous open-vocabulary text-to-motion generation methods

- MotionCLIP: Exposing Human Motion Generation to CLIP Space (ECCV 2022)



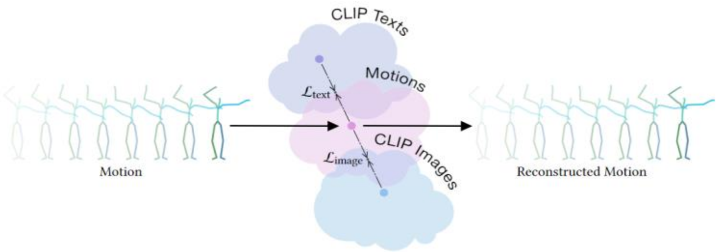
MotionCLIP Learns a motion VAE and regularize the latent space to close to CLIP text&image latent space:

- Offline generation
- Inputs for training and inference are different
- Requires paired text-motion training data

Tevet G, Gordon B, Hertz A, et al. Motionclip: Exposing human motion generation to clip space[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII. Cham: Springer Nature Switzerland, 2022: 358-374.

Previous open-vocabulary text-to-motion generation methods

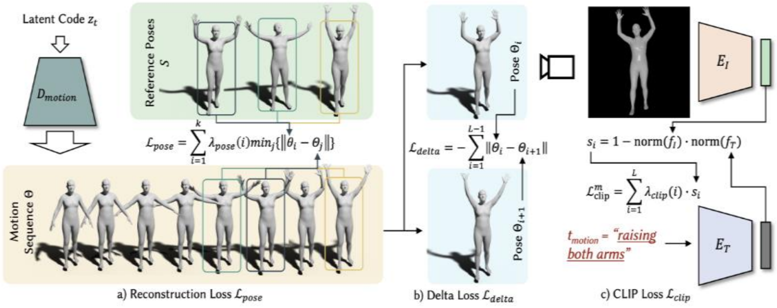
- MotionCLIP: Exposing Human Motion Generation to CLIP Space (ECCV 2022)



MotionCLIP Learns a motion VAE and regularize the latent space to close to CLIP text&image latent space:

- Offline generation
- Inputs for training and inference are different
- Requires paired text-motion training data

- AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars (TOG 2022)


Latent Code z_t
D_{motion}
Reference Poses S
Motion Sequence Θ
$\mathcal{L}_{pose} = \sum_{i=1}^{k} \lambda_{pose}(i) min_j\{\|\theta_i - \theta_j\|\}$
Pose Θ_i
Pose Θ_{i+1}
$\mathcal{L}_{delta} = -\sum_{i=1}^{L-1} \|\theta_i - \theta_{i+1}\|$
E_I
E_T
$s_i = 1 - \text{norm}(f_I) \cdot \text{norm}(f_T)$
$\mathcal{L}_{\text{clip}}^{m} = \sum_{i=1}^{L} \lambda_{clip}(i) \cdot s_i$
t_{motion} = "raising both arms"

a) Reconstruction Loss $\mathcal{L}_{pose}$

b) Delta Loss $\mathcal{L}_{delta}$

c) CLIP Loss $\mathcal{L}_{clip}$

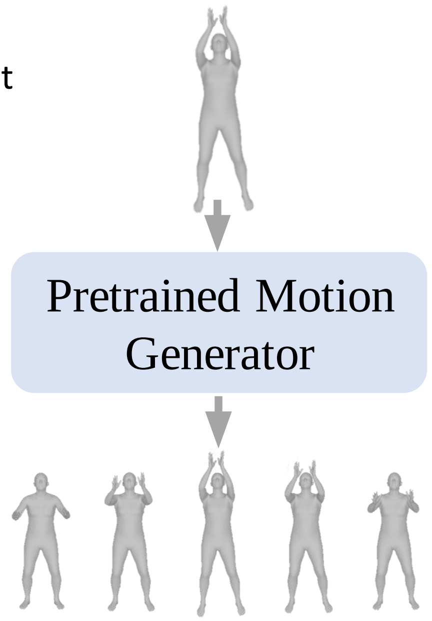
- The motion generation part of AvatarCLIP uses CLIP to match candidate poses of the given texts and use these poses to construct optimization loss
  - Zero-shot learning
  - Online matching and finetuning

Tevet G, Gordon B, Hertz A, et al. Motionclip: Exposing human motion generation to clip space[C]//Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII. Cham: Springer Nature Switzerland, 2022: 358-374.

Hong F, Zhang M, Pan L, et al. AvatarCLIP: zero-shot text-driven generation and animation of 3D avatars[J]. ACM Transactions on Graphics (TOG), 2022, 41(4): 1-19.

Offline Open-vocabulary Human Motion Generation (OOHMG)

- We propose a controllable and flexible text-to-motion generation method drawing the inspiration from prompt learning in NLP. To achieve this, our method includes
  - **A pose-prompt motion generator**
    - We pretrain a motion generator which is optimized to reconstruct the complete motion from the masked motion. And during inference, we can **reformulate the text into the masked motion as the generation prompt to synthesize the motion**.



Pretrained the Motion Generator

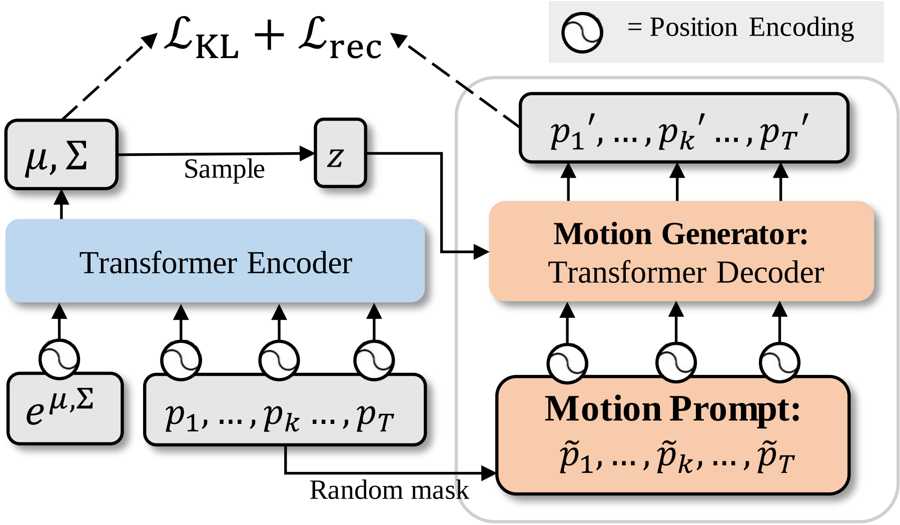
- In advanced language modeling in NLP, the language model learns to reconstruct the masked sentence from the randomly masked sentence in a self-supervised manner. Our motion generator also follows a similar training strategy.

Its loss function $\mathcal{L}_{\mathrm{m2m}}$ is as follows:

$$\mathcal{L}_{\mathrm{rec}}(p'_t, p_t) = \|p_t - p'_t\|_2 + \|v_t - v'_t\|_2,$$

$$\mathcal{L}_{\mathrm{m2m}}(m', m) = \textstyle\sum_{p'_t, p_t \in m', m} \mathcal{L}_{\mathrm{rec}}(p'_t, p_t) + \beta_{\mathrm{KL}} \times \mathcal{L}_{\mathrm{KL}},$$

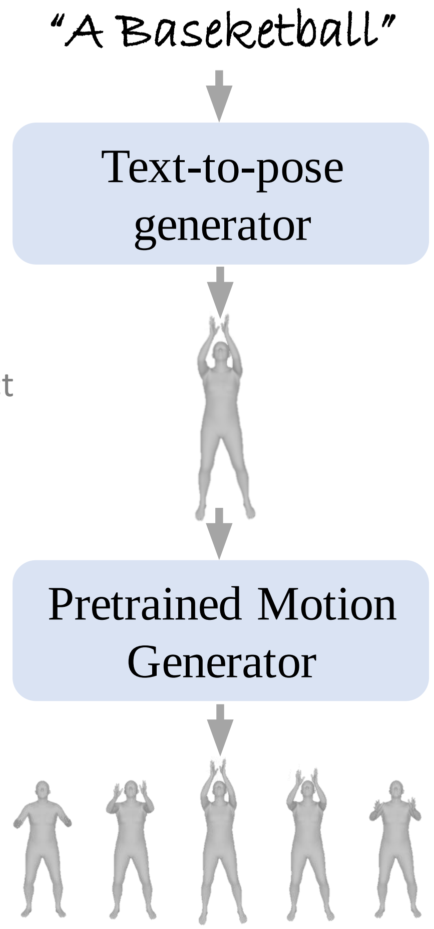
where m' is the reconstructed motion and m is the original motion.



$\mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{rec}}$
= Position Encoding
μ, Σ
Sample
z
$p_1', \dots, p_k' \dots, p_T'$
Transformer Encoder
Motion Generator:
Transformer Decoder
$e^{\mu,\Sigma}$
$p_1, \dots, p_k \dots, p_T$
Motion Prompt:
$\tilde{p}_1, \dots, \tilde{p}_k, \dots, \tilde{p}_T$
Random mask

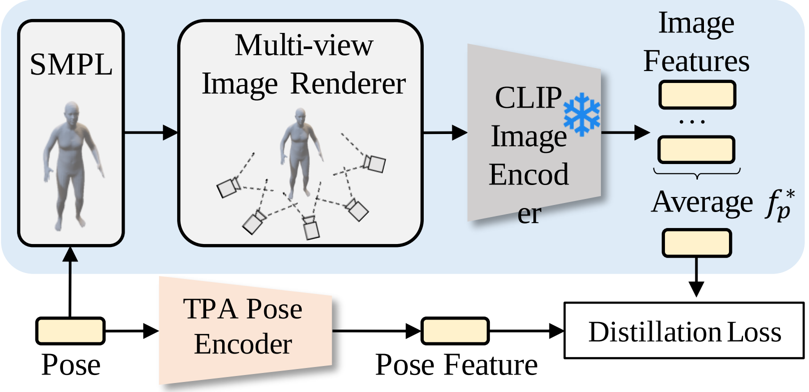
Offline Open-vocabulary Human Motion Generation (OOHMG)

- We propose a controllable and flexible text-to-motion generation method drawing the inspiration from prompt learning in NLP. To achieve this, our method includes
  - **A pose-prompt motion generator**
    - We pretrain a motion generator which is optimized to reconstruct the complete motion from the masked motion. And during inference, we can **reformulate the text into the masked motion as the generation prompt to synthesize the motion.**
  - **A generalized and parameterized text-to-pose generator**
    - To supervise the training process of the text-to-pose generator, we propose the first **text-3D pose alignment model, i.e., TPA.**
    - 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**.



Text-Pose Alignment Distillation

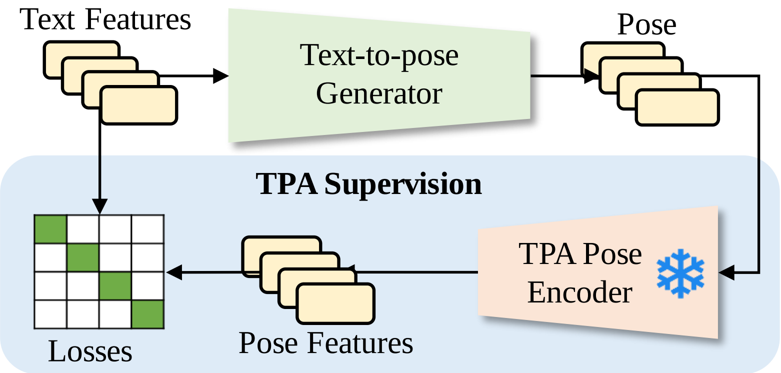
- We propose the first open-vocabulary text-3D pose alignment model, TPA.
- To leverage CLIP to align text and 3D pose, we can reuse the text encoder of CLIP and modify the image encoding process for the 3D pose. Following the same process as AvatarCLIP, **the pose feature is the averaged CLIP feature of the multi-view rendered images of the 3D pose.** However, this process is computation-heavy. And since the CLIP are trained on natural images, it's difficult for this process to optimize the 3D poses via backpropagation.
- To address these problems, we learn a small end-to-end neural network to distill this complex feature extraction process. By this means, we can limit the input domain of the alignment model to 3D poses, and manage to optimize the pose generator in an end-to-end learning manner.



$$\mathcal{L}(\mathrm{p};\mathrm{E_p}) = \left\|\mathrm{E_p}(p) - f_p^*\right\|_2 - \frac{\mathrm{E_p}(p)^T f_p^*}{|\mathrm{E_p}(p)||f_p^*|},$$

where $\mathrm{E_p}$ is the pose encoder of TPA, p is the input pose and f_p^* is the target pose feature.

Wordless training for Open-vocabulary Text-to-pose generator



The text-to-pose generator loss function $\mathcal{L}_{t2p}$ is：

$$\mathcal{L}_{TPA}(\boldsymbol{f_a}, \boldsymbol{f_b}) = -\sum_{i=1:B} \log \Pr(f_{a_i}|f_{b_i}) - log\Pr(f_{b_i}|f_{a_i}),$$

$$\mathcal{L}_{t2p}(\boldsymbol{p}^l, \boldsymbol{f}) = \mathcal{L}_{TPA}(E_p(\text{VPoser}(\boldsymbol{p}^l)), \boldsymbol{f}) + 0.1 \times \|\boldsymbol{p}^l\|_2,$$

where $\boldsymbol{p}^l$ is the output of the text-pose generator with $\boldsymbol{f}$ text feature. $\boldsymbol{p}^l$ is located in the latent space of VPoser (a pretrained pose VAE) and it can be decoded to obtain $\mathbf{p}$ as TPA's input.

- To generate poses for open-vocabulary texts, the text-to-pose generator should train with texts as diverse as possible.
- As the text space is combinatorial, it's impractical to enumerate all possible texts for training. It occurs to us that we can directly build a text-to-pose generator upon the normalized text feature space of CLIP instead of real-world text space.

Results

Text-to-motion Generation

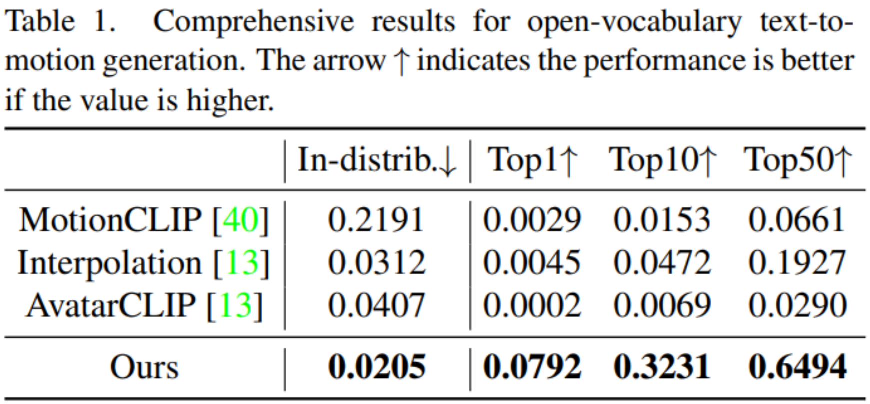


Table 1. Comprehensive results for open-vocabulary text-to-motion generation. The arrow ↑ indicates the performance is better if the value is higher.

	In-distrib.↓	Top1↑	Top10↑	Top50↑
MotionCLIP [40]	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**

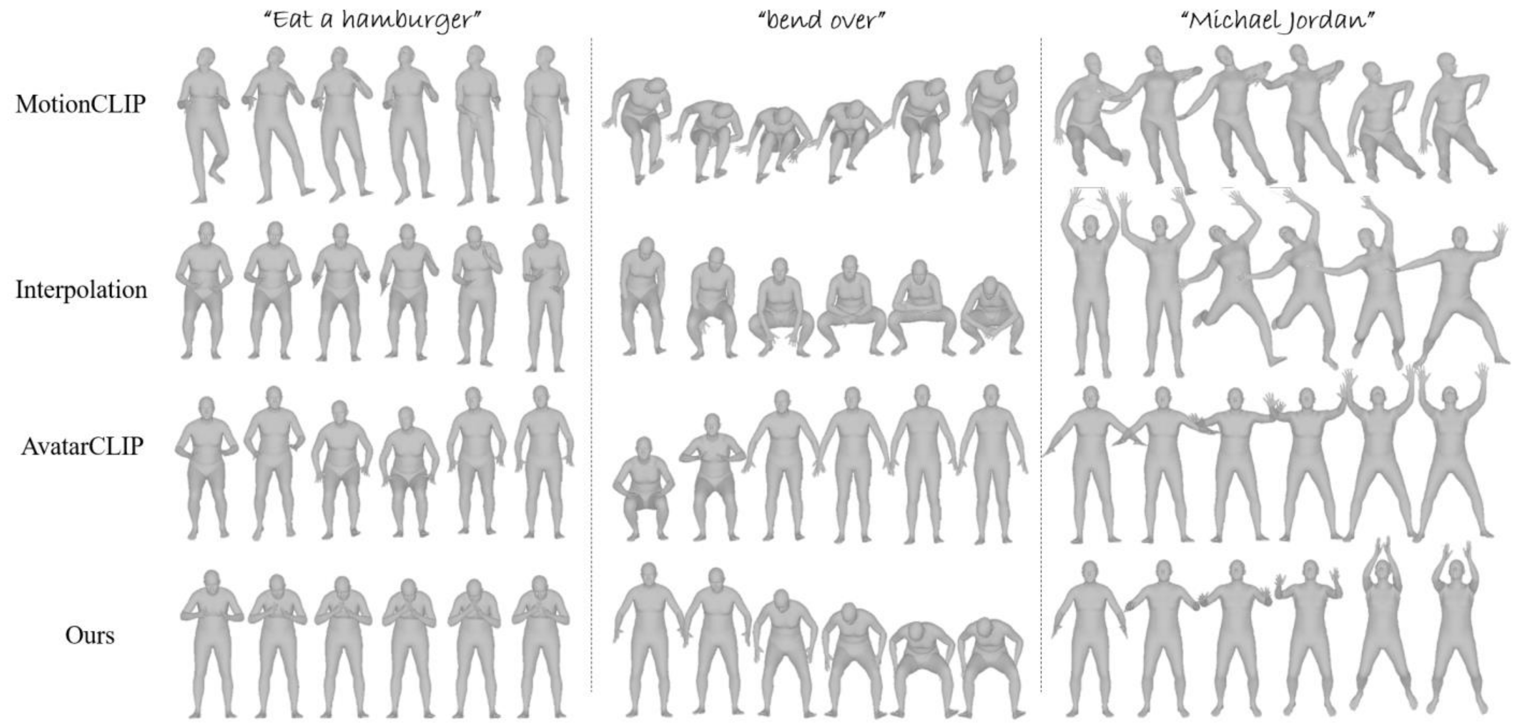


Figure 6. The visual comparison results of different open-vocabulary text-to-motion methods. The results are part of the generated motions due to the space limit.

Text-to-pose Generation

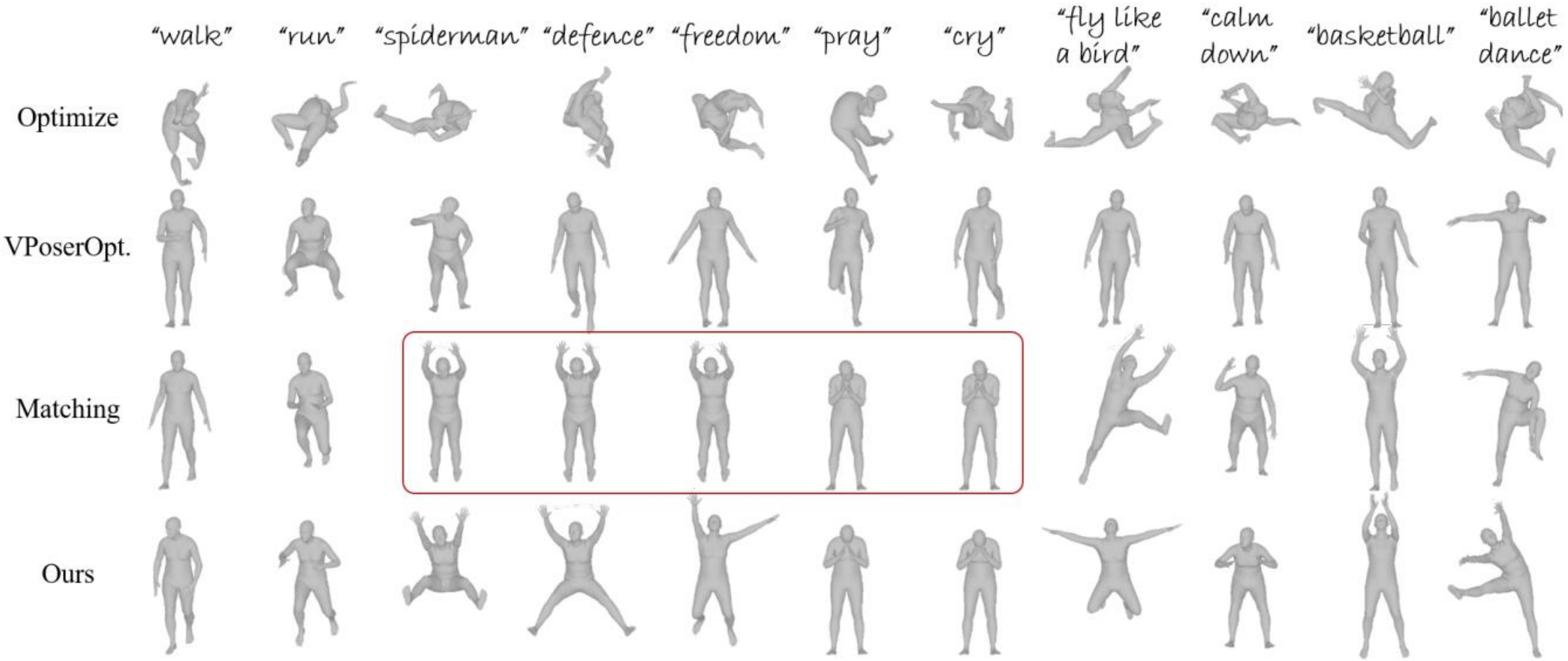
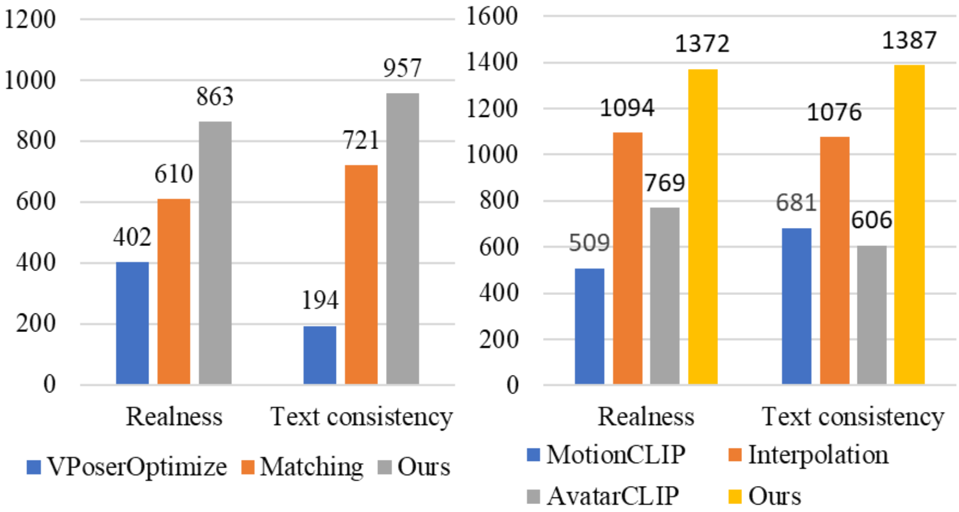


Figure 7. The visual results of different text-to-pose generation methods. Besides ours, the other baseline methods require online matching or optimizations.

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**

Human Evaluation
&
Inference Efficiency



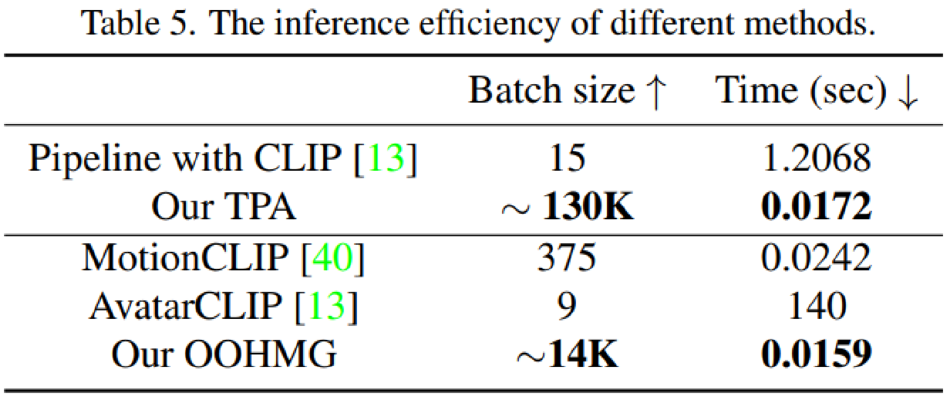


Table 5. The inference efficiency of different methods.

	Batch size ↑	Time (sec) ↓
Pipeline with CLIP [13]	15	1.2068
Our TPA	∼ **130K**	**0.0172**
MotionCLIP [40]	375	0.0242
AvatarCLIP [13]	9	140
Our OOHMG	∼**14K**	**0.0159**

Application example: Blender + OOHMG + Chat

Chat script(Blue text are the inputs to OOHMG)

Amy: "Hi"

Jackson waves hello

Jackson: “I have something to tell you”

Amy is curious

Jackson says that she has failed the exam

Amy is very sad and cried

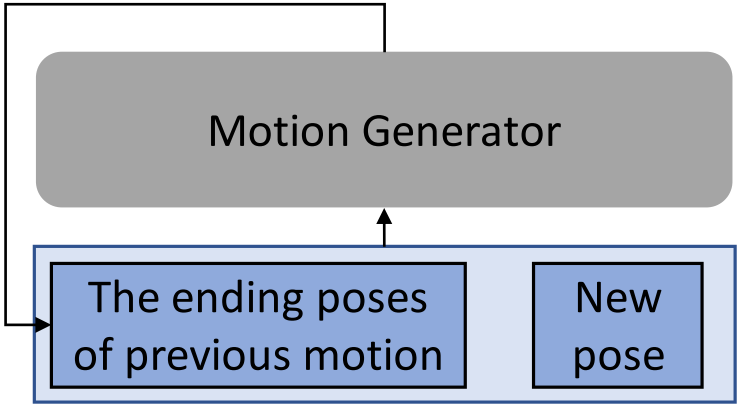
Jackson is worried

Amy covers her face and cries

Jackson apologizes

Amy suddenly surprises Jackson and says that she was pretending to cry

Amy and Jackson both laugh



Motion Generator

The ending poses of previous motion

New pose

Thank you for watching