

# Semantics-aware Motion Retargeting with Vision-Language Models

## Supplementary Material

In this document, we provide the following supplementary content:

- Dataset Details
- Evaluation Metrics
- Implementation Details
- Semantic Attention Visualization
- Ablation Study
- Additional Results

### A. Dataset Details

**Test set and fine-tuning set details.** We download 45 source motions of source characters including Y Bot, X Bot and Ortiz for testing and fine-tuning. More details can be found in Tab. 7. As we focus on the preservation of semantics, we choose the motions that are free of interpenetration and have obvious semantics which can be accurately described by text descriptions. To compute the MSE metric, we download the corresponding ground truth data provided by Mixamo of target characters including Aj, Kaya and Mousey. The ground truth is mainly created by copying the rotations of corresponding joints from the source character thus suffering from interpenetration and semantic loss. In order to equip our model with abundant information about geometry skinning and motion semantics, we carefully choose motions that explore the entire movement space of each source character for fine-tuning the model.

### B. Evaluation Metrics

We evaluate the performance of our method across three key dimensions: skeleton, geometry, and semantics. At the skeletal level, we measure the Mean Square Error (MSE) between retargeted joint positions  $\hat{\mathbf{P}}_B$  and ground truth  $\mathbf{P}_B$  provided by Mixamo, normalized by the character height  $h_B$ . We compare both the global and the local joint positions. The local MSE is calculated when the root position is aligned with the ground truth.

$$MSE = \frac{1}{h_B} \left\| \hat{\mathbf{P}}_B - \mathbf{P}_B \right\|_2^2 \quad (12)$$

At the geometric level, we evaluate the interpenetration percentage, which is calculated as the ratio of the number of penetrated vertices to the total number of vertices in each frame. A lower ratio indicates less interpenetration occurs.

$$PEN = \frac{\text{Number of penetrated vertices}}{\text{Total number of vertices}} \times 100\% \quad (13)$$

At the semantic level, we utilize the Image-Text Matching (ITM) score, Fréchet inception distance (FID) and semantics consistency loss as metrics to evaluate the semantics consistency. The task of Image-Text Matching [15] is to measure the visual-semantic similarity between an image and a textual description via a two-class linear classifier  $\mathcal{F}_c$  pre-trained in BLIP-2 [14]. To compute ITM, we first generate the textual description of the source motion with visual question answering and then compute the ITM score between the source textual description, denoted as *text*, and the rendered retargeted motion, denoted as *image*.

$$ITM = \mathcal{F}_c(\text{text}, \text{image}) \quad (14)$$

Fréchet inception distance (FID) is calculated between the semantic embedding distribution of retargeted motion and source motion. Let  $\mathcal{N}(\mu_s, \Sigma_s)$  denotes the source distribution, while  $\mathcal{N}(\mu_t, \Sigma_t)$  denotes the target distribution.

$$FID = \|\mu_s - \mu_t\|_2^2 + \text{tr}(\Sigma_s + \Sigma_t - 2(\Sigma_s^{\frac{1}{2}}\Sigma_t\Sigma_s^{\frac{1}{2}})^{\frac{1}{2}}) \quad (15)$$

### C. Implementation Details

**Training details.** We use four NVIDIA 3090Ti (24\*4GB) and the training process is divided into two stages. For skeleton-aware pre-training, the learning rate is set as 0.0003, the number of training epoch is set as 80 and the batch size is 16. For semantics fine-tuning, the learning rate is set to 0.0001 and batch size is 4. After 25 epoches, our model achieves state-of-the-art performance in motion retargeting and preserves the semantics of motion well. When fine-tuning our model with the interpenetration loss and the semantics consistency loss, we increase the weight of the interpenetration loss from 1.0 to 10.0 during the first 5 epochs. Because we observe that the performance of the vision-language model is unstable when there exists obvious interpenetration. And after 5 epoch, the weight goes back to 1.0. The initial hyper-parameters  $\lambda_r, \lambda_c, \lambda_a, \lambda_j, \lambda_p, \lambda_s$  for pre-training and fine-tuning loss functions are set to 10.0, 1.0, 0.1, 1.0, 1.0, 0.1. The vision language model we used is BLIP-2 [14] with pre-trained FlanT5-XXL [5] large language model and large scale pre-trained vision transformer. In order to generate more comprehensive text for our prompt, we use beam search with a beam width of 5. We also set the length-penalty to 1 which encourages longer answers.

**Network architecture.** The motion encoder and decoder architectures consist of three layers of graph convolutions. The first two layers utilize spatial graph convolution, adopting a message-passing scheme to aggregate features from

Method	MSE ↓	MSE <sup>lc</sup> ↓	Pen.% ↓	ITM ↑	FMD ↓	SCL ↓
SMT <sub>fwl</sub>	5.418	4.576	4.41	0.552	78.46	18.96
SMT <sub>fwq</sub>	0.739	0.517	4.56	0.658	2.497	0.191
SMT <sub>Ours</sub>	<b>0.284</b>	<b>0.229</b>	<b>3.50</b>	<b>0.680</b>	<b>0.436</b>	<b>0.143</b>

Table 4. Ablation study on semantic embedding. We compare the performance of the model fine-tuned with the image feature from CLIP [22] as semantic embedding (SMT<sub>fwl</sub>), the model fine-tuned with the features of the querying transformer as semantic embedding (SMT<sub>fwq</sub>) and the model fine-tuned with the features of the large language model encoder (SMT<sub>Ours</sub>)

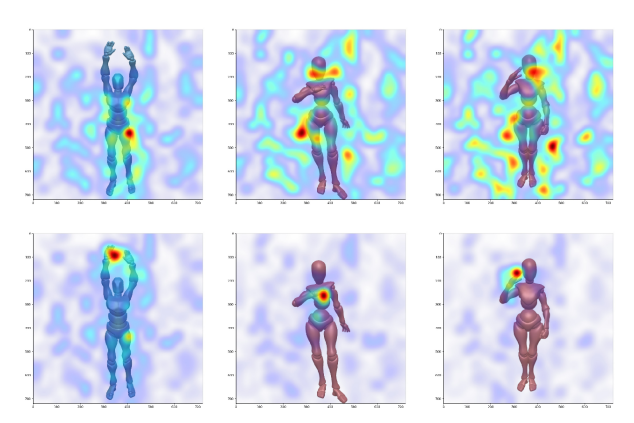


Figure 8. The cross attention map between question and image to generate semantic embedding, the first row is question without prompt and second row is question with prompt. Through guiding visual question answering, the semantic embedding concentrate on the localized regions which preserve motion semantics.

neighboring nodes as Eq. 16. The last layer is the temporal graph convolution, which maintains the same number of channels. The motion encoder receives joint rotations and positions as input and encodes them into latent motion embeddings, expanding the channels from 9 to 16 and 32. Subsequently, the motion decoder takes these latent motion embeddings as input and outputs the target joint rotations, gradually reducing the channels from 32 to 16 and 6. Additionally, the root joint positions are generated using a two-layer MLP, starting with node features from the root joint and expanding channels to 16 and 3.

$$\mathbf{x}_i' = \mathbf{x}_i + \sum_{j \in N(i)} g(\mathbf{W}_f[\mathbf{x}_i, \mathbf{x}_j, \mathbf{e}_{j,i}] + \mathbf{b}_f) \quad (16)$$

where  $\mathbf{x}_i$  is the feature of node  $i$ ,  $\mathbf{x}_i'$  is the updated feature of node  $i$ ,  $N(i)$  is the set of neighbor nodes of node  $i$ , and  $\mathbf{e}_{j,i}$  is the edge feature from node  $j$  to node  $i$ ,  $g$  is the LeakyReLU function,  $\mathbf{W}_f$  and  $\mathbf{b}_f$  are learnable parameters.

#### D. Semantic Attention Visualization

To gain insights into the preservation of motion semantics, we visualize the attention map between the question and

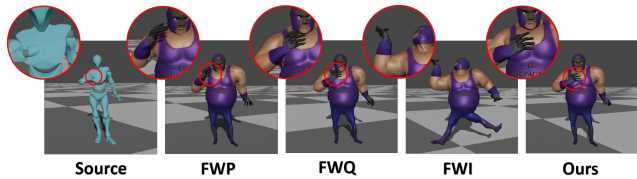


Figure 9. The qualitative comparison between the network fine-tuned without semantics (FWP), the network fine-tuned with the output of querying transformer (FWQ), the network fine-tuned with the output of image encoder (FWI), the network fine-tuned with the output of large language model encoder (Ours).

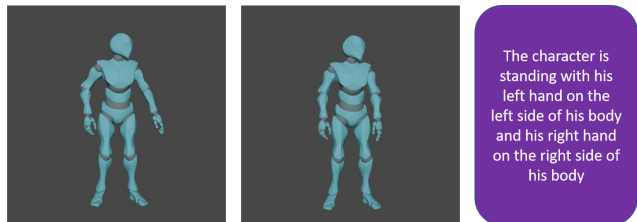


Figure 10. When the motion does not have obvious semantics, the text description can be the same while the semantic embedding from language encoder preserves the difference.

image. In Fig. 8, we illustrate how the semantic embedding accurately captures motion semantics in localized regions. This also clarifies the slight changes in the other joints of the character's skeleton with and without semantic consistency.

#### E. Ablation Study

**Latent semantic embedding.** We validate features from different levels: the output of the image encoder, the output of the querying transformer, the output of the large language model encoder. We visualize the features of three motions including Waving, Pointing and Salute with three source characters, including Y Bot, X Bot, Ortiz, using T-SNE [25] dimensionality reduction technique. The Fig. 11 shows that the features of the image encoder are clustered around characters rather than motions which indicates that the features contains more information on the appearance of the characters, while the features of the large language model encoder are clustered around motions and contains mainly motion semantics. We further use these features as semantic embedding to fine-tune our model and compute evaluation metrics in Tab. 4. The metrics and qualitative comparison indicates that appearance of character may lead to meaningless gradient for the model resulting in unnatural retargeted motion. Moreover, compared with the image features from CLIP, the features of the querying transformer and the large language model encoder focus more on relevant semantics with the help of guiding questions. Further more, We visualize the semantic embeddings outputted by VLM in Figure 12. The visualization indicates that VLM has captured rich motion semantics information, regardless

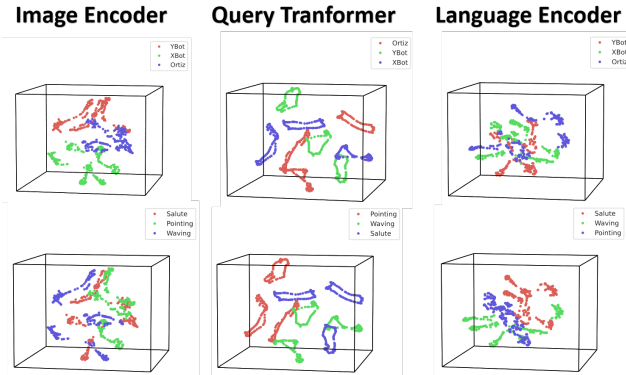


Figure 11. Features extracted from the image encoder, the query transformer and the encoder of the large language model visualized by T-SNE [25].

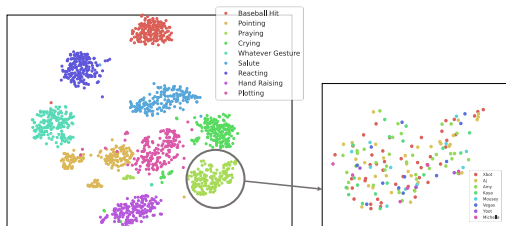


Figure 12. Visualization of semantic embedding space by t-SNE[25]. We visualize 9 motions of 8 characters with 72 sequences in total. The left figure demonstrates that frames of the same motion category are clustered. The right one shows that different characters in a single motion category are not clustered by characters.

of the appearance of the character, and is capable of guiding motion semantics preservation.

We also validate the possibility of using text descriptions as semantic embedding. But the text description of motion semantics is fuzzy and sparse. The Fig. 10 shows that during the transition phase of an action, the text descriptions remain the same. Comparing with text description, the latent features can provide dense supervision. Moreover, using text descriptions or output of decoder will bring more computation cost and higher non-linearity.

**Comparison with video semantics.** We compare the textual descriptions obtained from images and videos of motion sequences. The video-based vision-language model used for evaluation is SwinBert [17]. The Fig. 15 shows that although the video-based vision-language model can capture temporal information, the captured semantics is vague and lack details. The image-based vision-language model could generate more detailed and comprehensive descriptions and provide stronger supervision. Moreover, semantics of similar motions could be shared cross different motion sequence, which reduce the size of the fine-tuning set.

**Performance of different numbers of views.** We have conducted a new ablation in Table 5. The performance de-

Number of Views	MSE ↓	MSE <sup>LC</sup> ↓	Pen.% ↓	ITM ↑	FID ↓	SCL ↓
1 (front)	0.262	0.186	3.51	0.630	5.715	0.499
2 (left right diagonal)	0.274	0.195	3.49	0.651	2.849	0.277
3 (front, left, right)	0.284	0.229	3.50	0.680	0.436	0.143
5 (2+3)	0.286	0.233	3.50	0.681	0.433	0.141

Table 5. Quantitative comparison of different numbers of views.

Method	MSE ↓	MSE <sup>LC</sup> ↓	Pen.% ↓	ITM ↑	FID ↓	SCL ↓
NKN [26]	0.326	0.231	8.71	0.575	27.79	1.414
NKN [26] + VLM	0.392	0.308	4.44	0.665	2.687	0.223
SAN [2]	0.435	0.255	9.74	0.561	28.33	1.448
SAN [2] + VLM	0.481	0.339	5.08	0.659	2.798	0.258

Table 6. Quantitative comparison of different backbone networks.

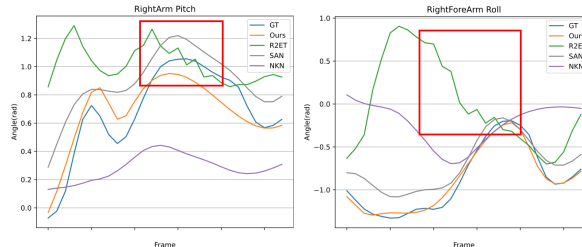


Figure 13. Example of the joint angle trajectory for the pitch and roll of the right arm.

creases to some extent with a single view due to depth information loss. Three perspectives, including the front, left and right view, can achieve fairly good results. Adding additional viewpoints after the third improves the outcome, but at a slower rate.

**Ablation study of skeleton network.** We have conducted a new ablation experiment in Table 6. The results show that the semantic module is applicable for different backbones and improves motion semantics preservation. The MSE metric has increased because the ground truth data in Mixamo dataset are not clean and suffer from interpenetration issues and semantic information loss [27].

## F. Additional Results

**More cases.** We provide additional cases to validate the effectiveness of the proposed method in the task of semantics-aware motion retargeting. Fig. 16 displays a gallery of retargeted results alongside their corresponding textual descriptions. Moreover, Fig. 17, Fig. 18, Fig. 19, and Fig. 20 present the retargeted motions of “Clapping”, “Crazy”, “React”, and “Fireball” from the source character to three different target characters. These qualitative results demonstrate that our method is able to produce high-quality motion retargeting results while preserving motion semantics.

**Retargeting motion from Internet videos.** We also conduct experiments on motion retargeting from wild videos on the Internet in Fig. 21. The human pose is estimated using the approach proposed in [20]. Then we perform motion retargeting from the human pose to Mixamo characters.

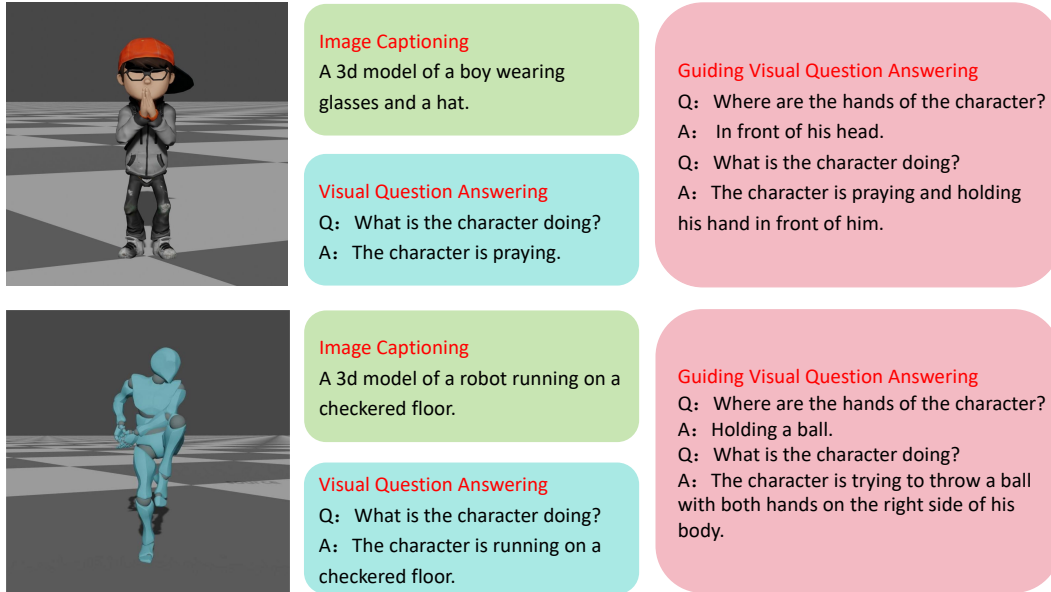


Figure 14. Text descriptions generated by different ways. The guiding visual question answering yields more comprehensive results.

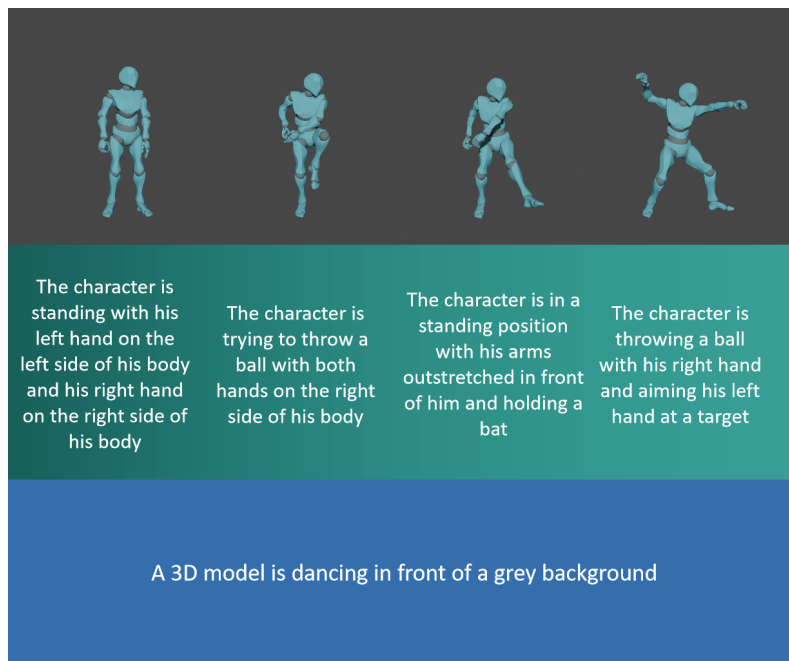


Figure 15. Text descriptions generated by the image language model and video language model. Green row is generated by image-based vision-language model. Blue row is the result of video-based vision-language model.

Considering the inherent errors in the human pose estimation, we extract the semantic embedding from the original video and apply the semantics consistency loss to further optimize the joint angles. We compare the results with and without optimization to validate the effectiveness of the semantics consistency loss. Despite the presence of some errors in human pose estimation, the retargeted motion accurately preserves the motion characteristics of the movement

in the original video.

**Smoothness.** We perform experiments to evaluate the smoothness of the retargeted motions. As an example, we visualize the joint angle trajectory for the pitch and roll of the right arm, and compare it with state-of-the-art methods. The Fig. 13 illustrates that our method delivers smoother motion compared to R2ET [29].

Index	Motion name	Search query	Source character	Length	Usage
1	Agreeing	Step Back Cautiously Agreeing	Y Bot	142	Test
2	Angry	Standing Angrily	Y Bot	576	Finetune
3	Baseball Hit	Baseball Base Hit	X Bot	118	Test
4	Baseball Pitching	Pitching A Baseball	Y Bot	119	Test
5	Cards	Dealing Cards	X Bot	274	Finetune
6	Charge	Point Onward Charge	Y Bot	172	Test
7	Clapping	Clap While Standing	Y Bot	36	Test
8	Counting	Counting To Five On One Hand	Y Bot	200	Test
9	Crying	Crying And Rubbing Eyes	X Bot	189	Test
10	Crazy Gesture	Crazy Hand Gesture	X Bot	151	Test
11	Defeat	Covering Face In Shame After Defeat	Y Bot	220	Finetune
12	Dismissing Gesture	Dismissing With Hand Forward	Y Bot	99	Test
13	Excited	Super Excited	X Bot	198	Finetune
14	Fireball	Street Fighter Hadouken	Y Bot	102	Test
15	Fist Pump	Pyping A Fist	Y Bot	115	Finetune
16	Focus	Shake Off Head Pain And Focus	X Bot	166	Finetune
17	Guitar Playing	Playing A Guitar	Y Bot	144	Test
18	Happy	Standing Happily	X Bot	301	Test
19	Hands Forward Gesture	Two Handed Forward Gesture	Ortiz	94	Test
20	Hand Raising	Raising A Hand	X Bot	123	Test
21	Insult	Insulting With Rude Gesture	X Bot	81	Test
22	Lead Jab	Long Body Jab	Y Bot	56	Test
23	Looking	Looking Off Into The Distance	Y Bot	241	Test
24	Loser	Showing Loser Gesture While Standing	Ortiz	99	Test
25	No	Indicating No	X Bot	151	Test
26	Padding	Padding A Single Oar Canoe	Y Bot	218	Finetune
27	Plotting	Evil Plotting	X Bot	100	Test
28	Pointing	Pointing While Seated	Ortiz	104	Test
29	Praying	Buckled Stand And Praying	Y Bot	36	Test
30	Reacting	Being Surprised And Looking Right	Ortiz	111	Test
31	Salute	Formal Military Salute	X Bot	86	Test
32	Shaking Hands 2	2 People Shaking Hands Part 2 - Male	Y Bot	132	Test
33	Smoking	Idle Smoking	Y Bot	538	Test
34	Standing Greeting	Greeting While Standing	Ortiz	154	Test
35	Thankful	Being Thankful While Standing	X Bot	91	Test
36	Taunt	Taunting Pointing At Wrist	X Bot	86	Test
37	Taunt Gesture	Taunt Gesture	Ortiz	60	Finetune
38	Talking	Asking A Question	X Bot	156	Finetune
39	Talking	Male Talking On The Cell Phone	Y Bot	145	Finetune
40	Telling A Secret	Telling A Secret	Ortiz	328	Finetune
41	Victory	Celebrating After A Win While Seated	Ortiz	184	Finetune
42	Waving	Waving With Both Hands	Ortiz	96	Finetune
43	Whatever Gesture	Whatever Gesture	Ortiz	46	Test
44	Yawn	Big Yawn While Standing	X Bot	251	Finetune
45	Yelling	Yelling In Anger	Ortiz	236	Finetune

Table 7. 45 source motion sequences of 3 characters for testing and fine-tuning.


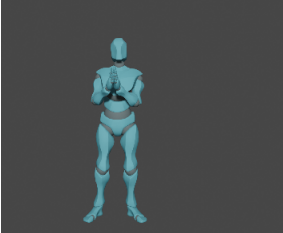

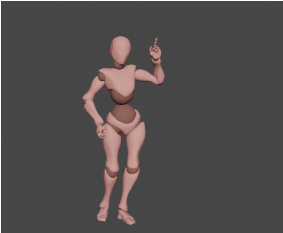


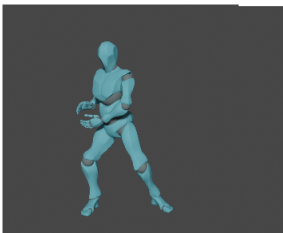

SOURCE	TARGET1	TARGET2	TARGET3	DESCRIPTION
				The character is standing with his hands positioned close together in front of his chest
				The character is praying and holding his hand up in front of him
				The character is pointing upwards with his left hand and putting his right hand on the hips
				The character is looking off to his right side and raising both hand in front of his chest
				The character is holding a ball with one hand near the chest and one hand near the raised leg

Figure 16. Snapshots of motions retargeted from the source character to three different characters and corresponding textual descriptions.

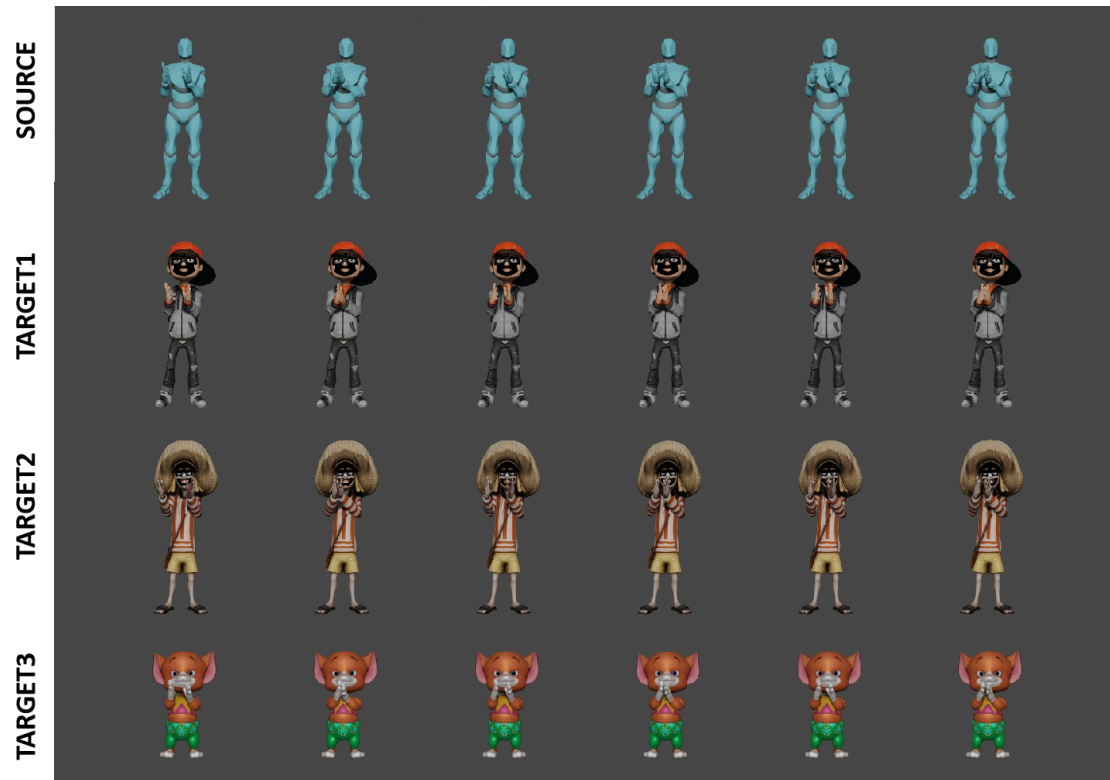


Figure 17. Snapshots of motion sequence “Clapping” retargeted from the source character to three different characters.

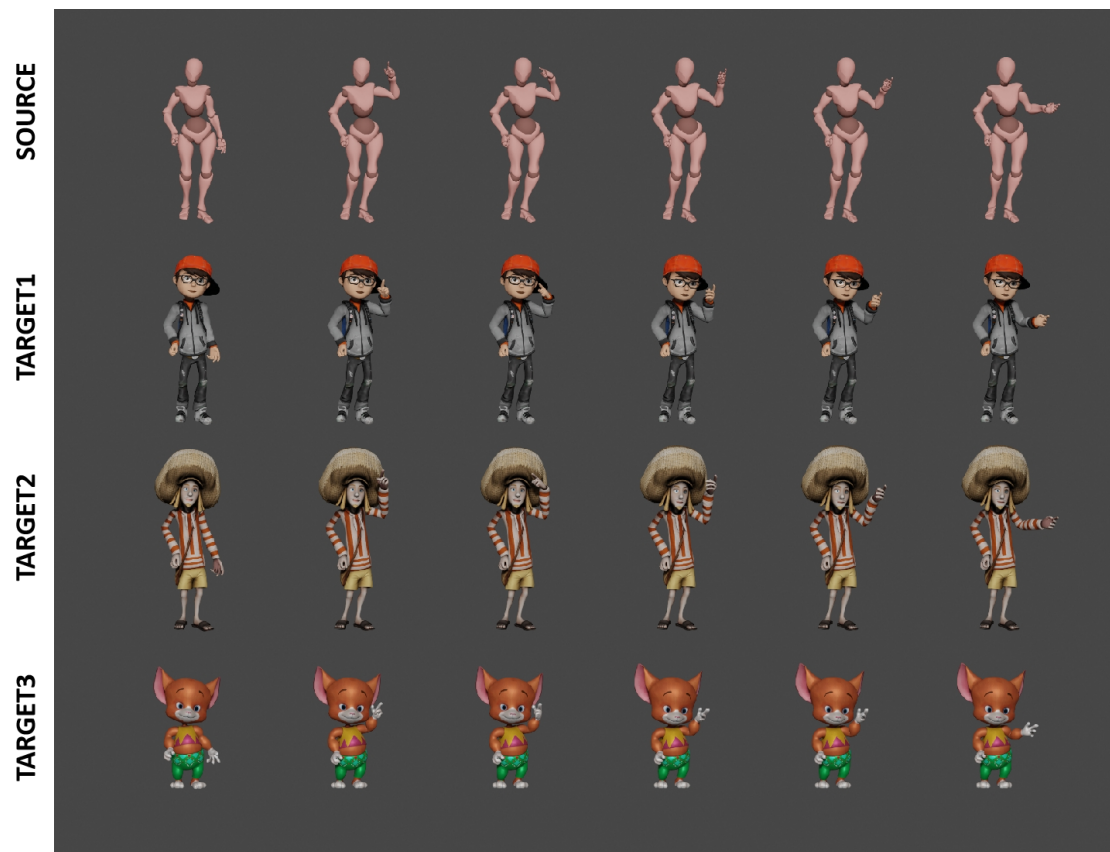


Figure 18. Snapshots of motion sequence “Crazy” retargeted from the source character to three different characters.

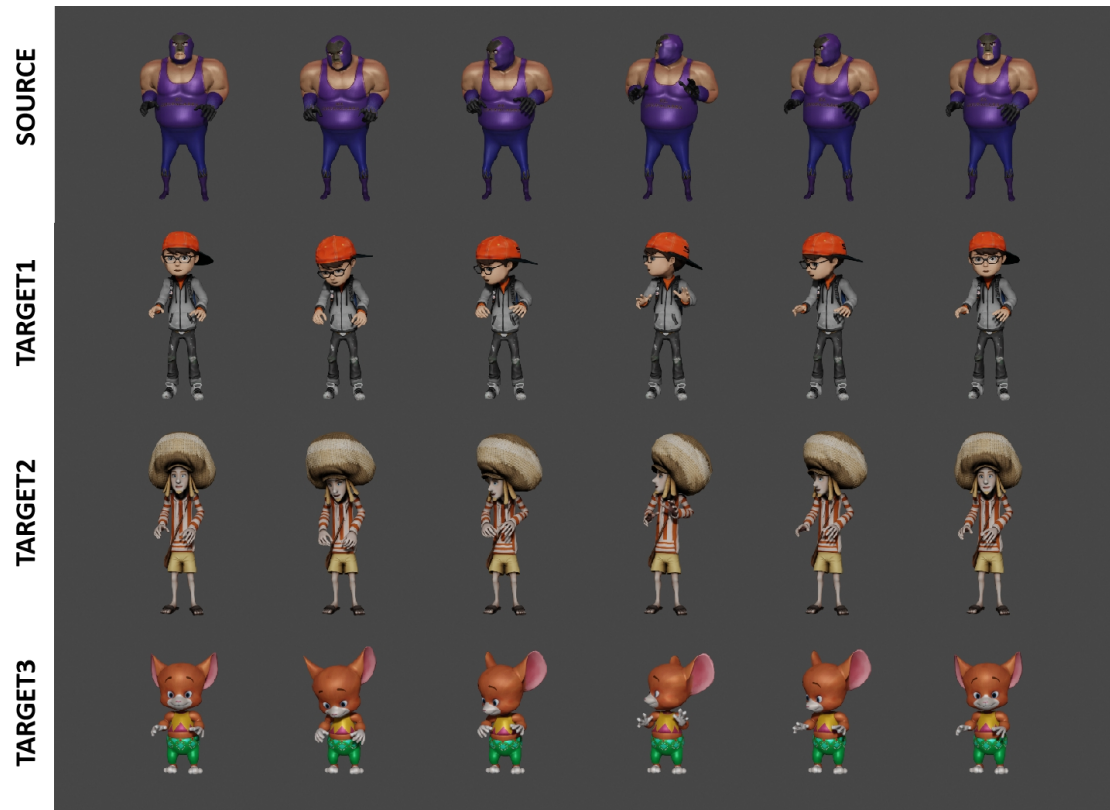


Figure 19. Snapshots of motion sequence “React” retargeted from the source character to three different characters.

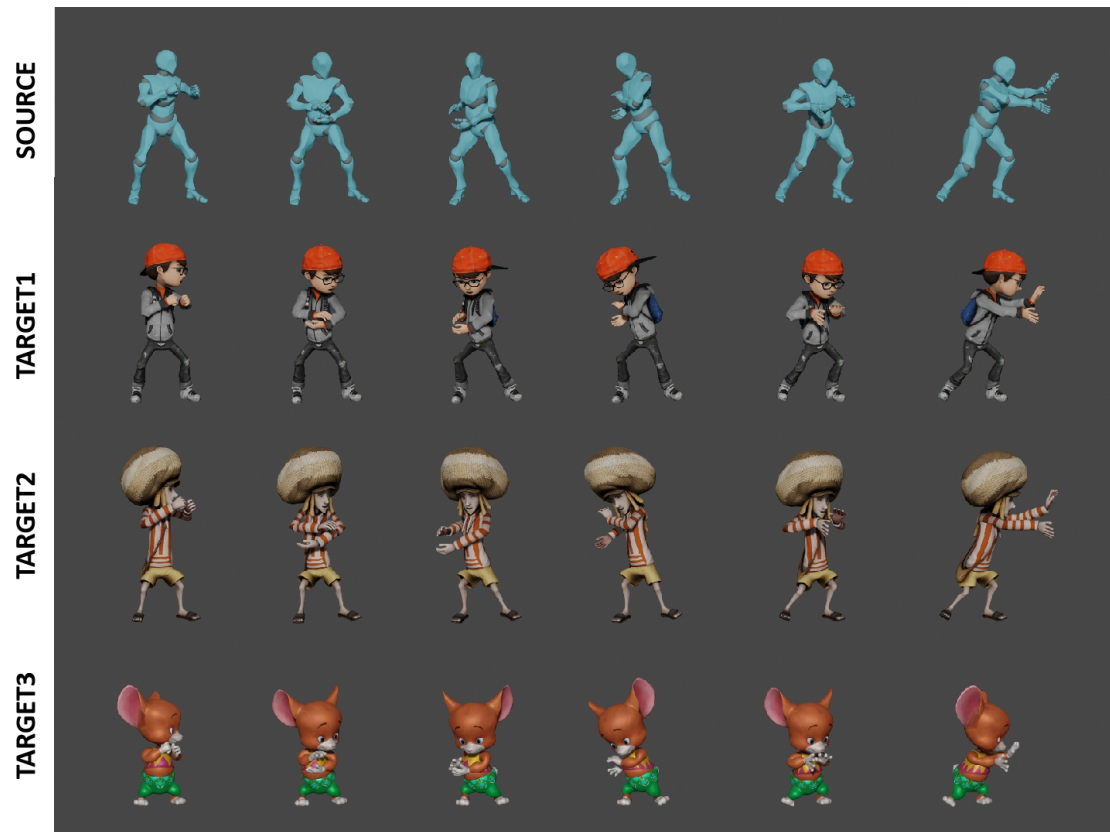


Figure 20. Snapshots of motion sequence “Fireball” retargeted from the source character to three different characters.



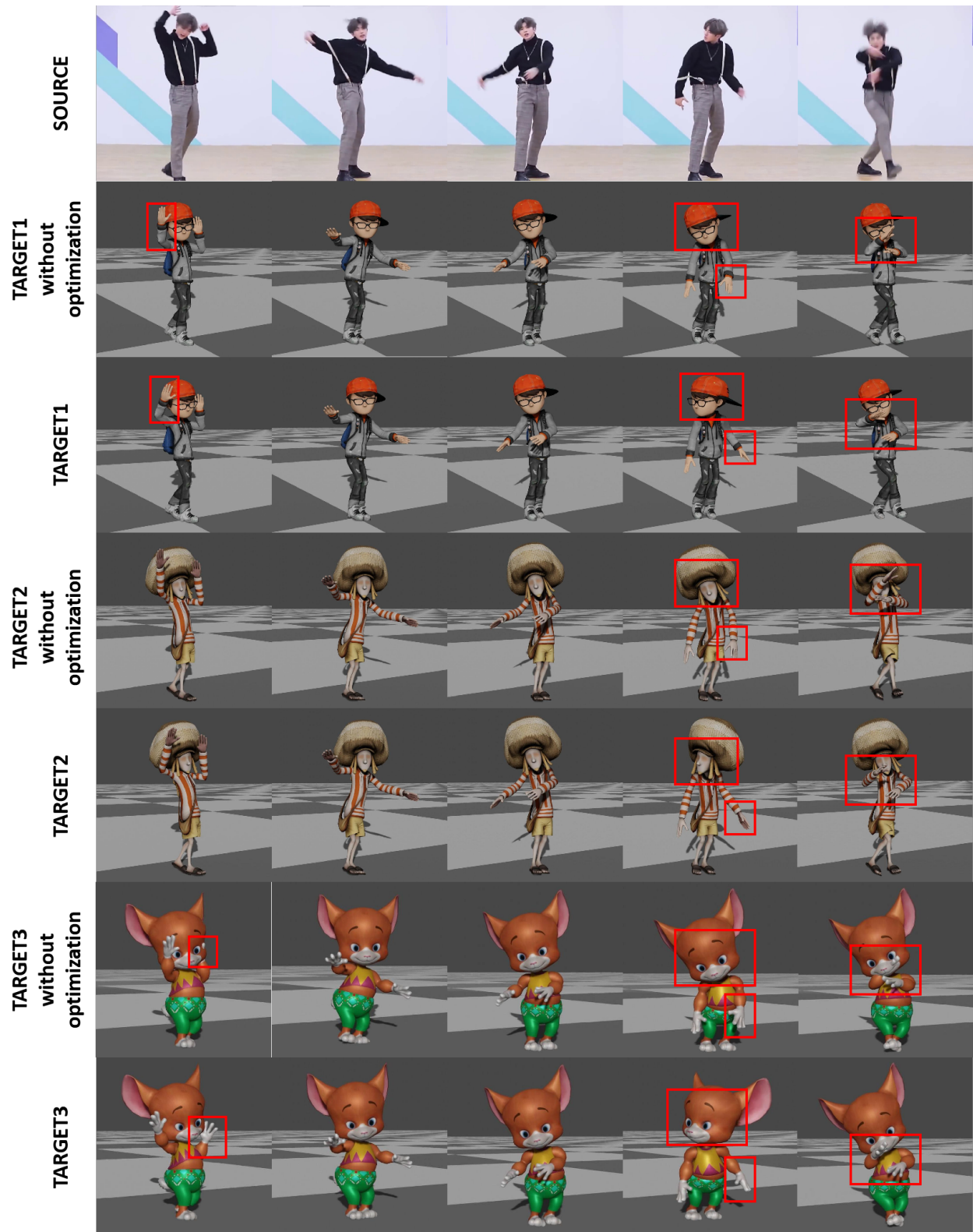


Figure 21. Snapshots of motions retargeted from the wild video on the Internet to three different characters with and without optimization.