Supplementary Material

Longyun Liao
McMaster University
liaol13@mcmaster.ca

Rong Zheng
McMaster University
rzheng@mcmaster.ca

1. Additional Architecture Details

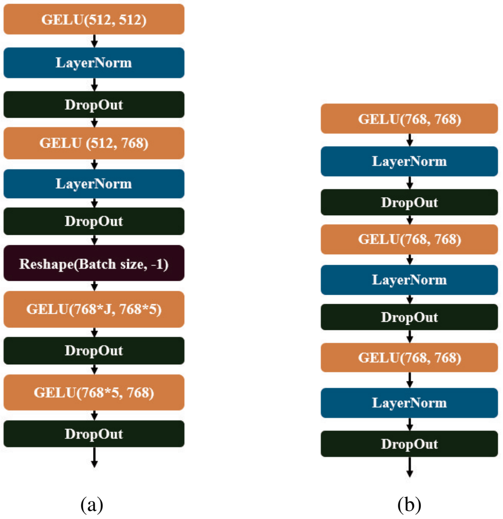

Figure 1. The architecture of pose embedding pooling layers (a) and text embedding pooling layers (b), where J represents the number of joints.

In this supplementary material, we provide a detailed illustration of the architecture for the Text Embedding and Pose Embedding Pooling Layers. As shown in Fig. 1, all linear layers are activated by GELUs [4], followed by a layer normalization layer and/or a dropout layer. For the Pose Embedding Pooling layers, due to the extra joint dimension, we first reshape the input in the middle, then apply a weighted summation over the last dimension.

2. Experimental Details

2.1. Configuration

The network is implemented in PyTorch and trained on a single NVIDIA H100 GPU for 300 epochs using an AdamW optimizer, with a learning rate of 0.0005 and a batch size of 16.

2.2. Pretraining

For the pretraining datasets BABEL [10] and AMASS [7], we first render the SMPL+H [11] parametric model, then extract the 3D keypoints using a regression matrix [1]. The original AMASS dataset, which has a frequency of 120 Hz, is resampled to 30 Hz. In 50% of training instances, we apply *random masking*: 5% of joints are masked at the joint level and 15% of frames are masked at the frame level. In 25% of instances, we use *time-window masking* with the window length uniformly sampled between $T_1 = 30$ and $T_2 = 80$ frames. The remaining 25% of instances use *body-part masking*. Additionally, we set the maximum text window to 20 tokens and the maximum motion window to 243 frames.

During pretraining, for the contrastive loss $\mathcal{L}_{con}$, the temperature is set to 0.1, and each pair of pose and text embeddings is compared against 16 negative samples. The loss weights are set to $\lambda_{con} = 0.5$, $\lambda_{3D} = 1$, and $\lambda_v = 20$, and the model is trained for 300 epochs.

3. Qualitative Results

3.1. Qualitative Results on the Pose Encoder's Semantic Embedding

In this section, we present additional qualitative results to assess the semantic embedding capability of the fine-tuned pose encoder on the Human3.6M dataset [5]. Specifically, we compare the similarity scores between pose embeddings and text embeddings of two selected action labels, followed by applying the SoftMax function to these scores. As shown in Fig. 3, the pose encoder can distinguish not only between semantically similar actions (e.g., bend down and kneel) and semantically related actions (e.g., walk and stand), but also semantically inverse actions (e.g., sit down vs. get up from a chair, walk forward vs. walk backward).

4. Motivation and Justification for the Modified InfoNCE Loss

We compare the *original* InfoNCE (cross-entropy with hard targets) to our *modified* InfoNCE that combines **focal loss**

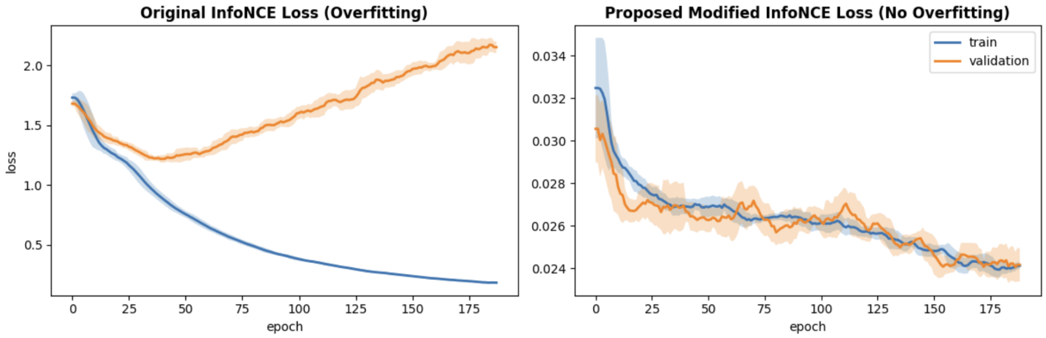

Figure 2. Two-panel comparison under identical training setup. **Left:** *Original InfoNCE* shows overfitting: training loss keeps decreasing while validation loss degrades after around epoch 40. **Right:** *Proposed modified InfoNCE* avoids overfitting and yields stable or improved validation performance.

Table 1. Ablation study comparing the proposed modified InfoNCE loss with the original InfoNCE formulation on MPI-INF-3DHP and Human3.6M. MMM denotes masked motion modeling. Results are reported using P1 and P3 metrics (mm).

Method (Loss)	MPI-INF-3DHP (P1)	Human3.6M (P1)	Human3.6M (P3)
Baseline (w/o InfoNCE)	16.7	37.5	49.0
Baseline + original InfoNCE + MMM	16.3	37.5	48.5
Baseline + MMM + **Modified InfoNCE Loss (ours)**	**15.5**	**36.7**	**47.0**

with **KL divergence** under the same architecture, data, optimizer, and schedule. As shown in Fig. 2, the original loss reduces training error yet degrades validation performance (left), indicating that the learned semantic alignment is not sufficiently generalizable. This observation matches our motivation: motion–language datasets [6, 10] are **highly imbalanced**, and human action recognition is **inherently non-deterministic** (motions and action labels lack a one-to-one correspondence).

5. Additional Statistics

5.1. Visibility Estimation

To quantitatively assess the extent of occlusion in the Human3.6M dataset [5], we computed a hard visibility score on the test set using a finetuned Stacked Hourglass network [8] as our 2D keypoint detector. For each joint at every frame, a joint was counted as visible if its detection confidence exceeded a threshold of 0.6, and occluded otherwise. The per-action visibility was obtained by averaging the proportion of visible joints across all frames belonging to that action in the test set. This procedure yields a score between 0 and 1, where higher values indicate that a larger fraction of joints are consistently visible. Table 2 summarizes the per-action hard visibility statistics on the Human3.6M test set.

5.2. Measuring Distributional Discrepancy across Datasets

Distribution distance for motion datasets (MMD). We quantify the distributional discrepancy between two motion datasets (e.g., sets of 3D joint trajectories) using the Maximum Mean Discrepancy (MMD). Given feature vectors extracted from each motion sample (e.g., root-centered, standardized per-frame statistics or learned embeddings), MMD measures the RKHS distance between the kernel mean embeddings of the two empirical distributions. With any positive definite kernel k, $\mathrm{MMD}(P,Q) \geq 0$ and equals 0 iff $P = Q$. We adopt an RBF kernel and report (squared) MMD; smaller values indicate closer motion distributions.

$$\begin{aligned}\mathrm{MMD}^2(P,Q;k) = \mathbb{E}_{X,X'\sim P}\, k(X,X') + \mathbb{E}_{Y,Y'\sim Q}\, k(Y,Y') \\ - 2\,\mathbb{E}_{X\sim P,\, Y\sim Q}\, k(X,Y). \qquad (1)\end{aligned}$$

$$\begin{aligned}\widehat{\mathrm{MMD}}^2 = \frac{1}{n(n-1)}\sum_{i<j} k(x_i,x_j) + \frac{1}{m(m-1)}\sum_{i<j} k(y_i,y_j) \\ - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k(x_i,y_j). \qquad (2)\end{aligned}$$

$$k(x,y) = \exp\Big(-\tfrac{\|x-y\|_2^2}{2\sigma^2}\Big). \qquad (3)$$

Motion feature construction (brief). We convert each clip to root-centered 3D joint coordinates (subtracting the pelvis per frame) and apply z-score standardization. From the standardized sequence we compute lightweight per-frame descriptors—mean/std across joints of radial magnitude, joint speed, and acceleration (via finite differences)—and concatenate them into compact features used for MMD.

Table 2. Human3.6M per-action hard visibility ($\theta = 0.6$). Higher is better.

Action	Directions	Discussion	Eating	Greeting	Phoning	Posing	Purchases	**Sitting**
Visibility	0.947	0.930	0.943	0.952	0.914	0.969	0.921	**0.899**
Action	**SittingDown**	Smoking	**Photo**	Waiting	Walking	WalkingDog	WalkingTogether	mean
Visibility	**0.830**	0.915	**0.908**	0.939	0.958	0.919	0.972	0.928

Table 3. Per-action (squared) MMD (RBF). Left: **H3.6M** validation → **H3.6M** training; Right: **H3.6M** validation → **BABEL** training. Lower MMD is better; larger improvement (mm) is better.

Action	$\mathbf{MMD^2}$ (v→t)	$\mathbf{MMD^2}$ (v→BABEL)	**Improvement (mm)** ↑
Directions	0.015	0.246	0.5
Discussion	0.065	0.298	0.0
Eating	0.019	0.189	0.5
Greeting	0.017	0.442	0.4
Phoning	0.036	0.127	0.2
Photo	0.026	0.154	0.9
Posing	0.005	0.337	-0.4
Purchases	0.013	0.197	1.9
Sitting	0.037	0.152	0.7
SittingDown	0.014	0.233	4.6
Smoking	0.012	0.157	0.5
Waiting	0.024	0.326	0.3
Walking	0.073	0.434	0.5
WalkDog	0.010	0.321	1.4
WalkTogether	0.047	0.462	0.2

v: Human3.6M validation; t: Human3.6M training. RBF bandwidth via median heuristic.

Improvement over the baseline. Our baseline is *MotionBERT* [14], pretrained only on motion data without semantic information. Our method augments pretraining with semantic signals (e.g., action/text annotations) via an additional alignment objective, while keeping the downstream evaluation protocol identical. For each action a on the H3.6M validation split, we report the improvement as the reduction in MPJPE (mm) relative to the baseline:

$$\Delta_a = \mathrm{MPJPE}_{\text{baseline},a} - \mathrm{MPJPE}_{\text{ours},a} \text{ (mm)}, \quad (4)$$

so larger Δ_a indicates larger performance improvement.

Distribution matching. For each action, we quantify distributional discrepancy via (squared) MMD on root-centered, standardized motion features. The motion features are obtained by root-relative position with normalization. We find that some actions with smaller MMD between **H3.6M** validation and **BABEL** training tend to benefit more from our method, while the MMD between **H3.6M** validation and **H3.6M** training does not exhibit a clear relationship with the improvement. This pattern indicates that our improvements primarily stem from the added *semantic* information rather than distribution matching alone; notably, actions with larger gains are *not* simply those under-represented in the H3.6M training set, as the MMD between validation and H3.6M training shows no consistent association with the gains. (Note that the baseline *MotionBERT* is pretrained on a motion corpus that *strictly contains* the motion-only subset we use for *LangPose*. Our *LangPose* further augments its subset with textual (semantic) supervision.)

5.3. Inference efficiency

At inference, only the motion encoder is used; it contains approximately 42.4M trainable parameters.

6. Motion Captioning

For motion-to-text captioning, we follow the TM2T framework [3] and replace its motion encoder with ours after the pretraining stage. Notably, we found out that a vanilla motion encoder—without our pretraining and without TM2T's motion tokenization—fails to learn the captioning task effectively during fine-tuning.

Unlike TM2T, which first discretizes motion with a VQ-VAE [12] and feeds motion tokens to the encoder, our model directly consumes skeleton joint positions $X \in \mathbb{R}^{T \times J \times 3}$ (root-relative and normalized). All other components (text decoder, training schedule, and losses) are kept the same as TM2T to isolate the effect of the motion representation.

As shown in Tab. 4, on the KIT-ML test set [9], TM2T surpasses traditional sequence models (e.g., RAEs [13], SeqGAN [2]) on all reported metrics (R-Precision, BLEU@1/4, ROUGE, CIDEr), underscoring its advantage over earlier baselines. Our method markedly outperforms traditional approaches and achieves performance comparable to TM2T on several metrics, highlighting its promise for future motion-captioning applications.

References

[1] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Computer Vision – ECCV 2016*. Springer International Publishing, 2016. 1

[2] Yusuke Goutsu and Tetsunari Inamura. Linguistic descriptions of human motion with generative adversarial seq2seq learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4281–4287, 2021. 3, 4

Table 4. Quantitative evaluation on the **KIT-ML** test set. Higher is better for R Precision, BLEU, ROUGE, and CIDEr

Methods	R Precision↑			BLEU@1↑	BLEU@4↑	ROUGE↑	CIDEr↑
	Top 1	Top 2	Top 3				
RAEs [13]	0.034	0.063	0.106	30.6	0.10	25.7	8.00
SeqGAN [2]	0.109	0.345	0.425	3.12	5.20	32.4	29.5
TM2T [3]	0.359	0.561	0.668	46.7	18.4	44.2	79.5
Ours	0.308	0.493	0.612	31.6	0.23	31.5	23.0

[3] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *ECCV*, 2022. 3, 4

[4] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. 1

[5] 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*, 36(7):1325–1339, 2014. 1, 2

[6] Franziska Krebs, Andre Meixner, Isabel Patzer, and Tamim Asfour. The kit bimanual manipulation dataset. In *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, pages 499–506, 2021. 2

[7] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, 2019. 1

[8] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Computer Vision – ECCV 2016*, pages 483–499, Cham, 2016. Springer International Publishing. 2

[9] Matthias Plappert, Christian Mandery, and Tamim Asfour. The KIT motion-language dataset. *Big Data*, 4(4):236–252, 2016. 3

[10] Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 722–731, 2021. 1, 2

[11] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Trans. Graph.*, 36(6), 2017. 1

[12] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc. 3

[13] Tatsuro Yamada, Hiroyuki Matsunaga, and Tetsuya Ogata. Paired recurrent autoencoders for bidirectional translation between robot actions and linguistic descriptions. *IEEE Robotics and Automation Letters*, 3(4):3441–3448, 2018. 3, 4

[14] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 3

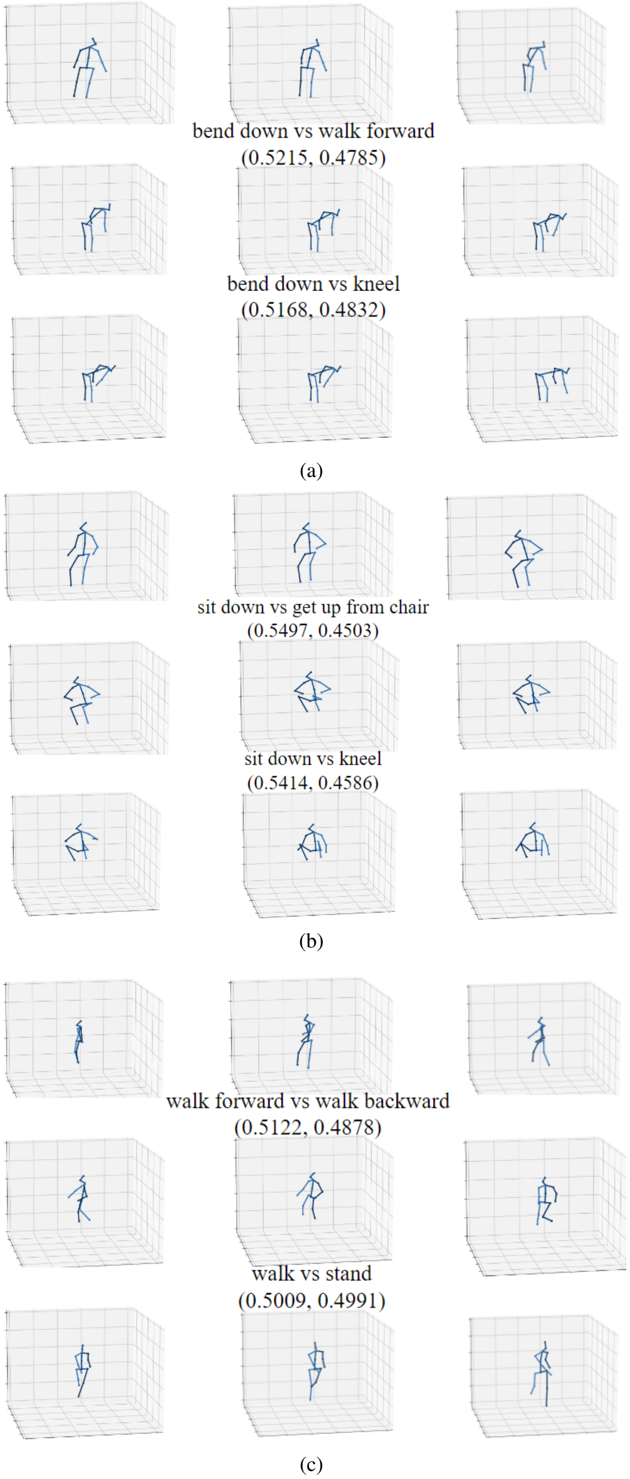

Figure 3. Qualitative Results on the Pose Encoder's Semantic Embedding. Each clip is manually selected from the Human3.6M dataset: (a) and (c) are from the activity 'Walking Dog', while (b) is from 'Sitting Down'.