PoseEmbroider: Towards a 3D, Visual, Semantic-aware Human Pose Representation – *Supplementary Material* –

Ginger Delmas[1,2], Philippe Weinzaepfel[2]
Francesc Moreno-Noguer[1], and Grégory Rogez[2]

[1] Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
[2] NAVER LABS Europe

https://europe.naverlabs.com/research/PoseEmbroider/

In this supplementary material, we present examples of tri-modal data samples from BEDLAM-Script and BEDLAM-Fix (Section A), and complete the explanations given in the main paper about these augmentations of BEDLAM (*e.g.* image selection criteria). We provide additional qualitative results and analysis (including limitations) of our models in Section B. The original annotations associated to the queries presented in the main paper are available in Section C. Finally, we give implementation details in Section D and discuss responsibility to human subjects (Section E).

A More about BEDLAM-Script and BEDLAM-Fix

Dataset examples. Figure A1 shows some examples of tri-modal samples from BEDLAM-Script. Figure A2 does the same for BEDLAM-Fix.

Additional details about data selection. To select samples from BEDLAM, we begin by a filtering of the images based on joint visibility, then proceed with the farthest pose sampling algorithm described in the main text of this paper. Here is the list of criteria used to select people images:

- At least 16 of the main body joints had to be within the image boundaries (although they could be subject to occlusion).
- The person was required to be at the forefront (*i.e.*, positioned closest to the front compared to other individuals in the same image). However, we relaxed this condition slightly by also considering individuals positioned further in the background, provided that at least 70% of their bounding box did not overlap with the bounding box of someone positioned closer to the front.
- At least one side of the human bounding box (upscaled by a factor 1.1) had to be more than 224 pixels.

Paired elements (poses, images) in BEDLAM-Fix are part of BEDLAM-Script.

Fig. A1: Examples from BEDLAM-Script. A text describes the 3D pose represented in the image.

B Additional qualitative results

B.1 Text instruction generation

The automatic generation of pose instructions has several applications. It makes it possible to give correctional feedback to a trainee by comparing their pose to a trainer's. It also allows automatic narration to accompany sport training videos. In Figure A3, we provide qualitative results on real-world images depicting pilates moves. These pictures were obtained from YouTube videos of pilates classes.

We notice that the text instructions globally fit the different situations, especially for sitting and standing poses (four last columns of the first row). In particular, we find that the model is able to distinguish a set of fine-grained body changes, about arm position (1st row, columns 2,3 and 5), leg or arm bending (1st row, column 4; 2nd row, columns 1 and 3), body twist (2nd row, columns 2, 4 and 5), head rotation (1st row, middle example) and so forth.

However, there are still small perception mistakes (*e.g. first image of the first row*: the right foot is not exactly on the floor; *middle image of the first row*: the elbows should not really be bent; *fourth image of the first row*: the right hand is

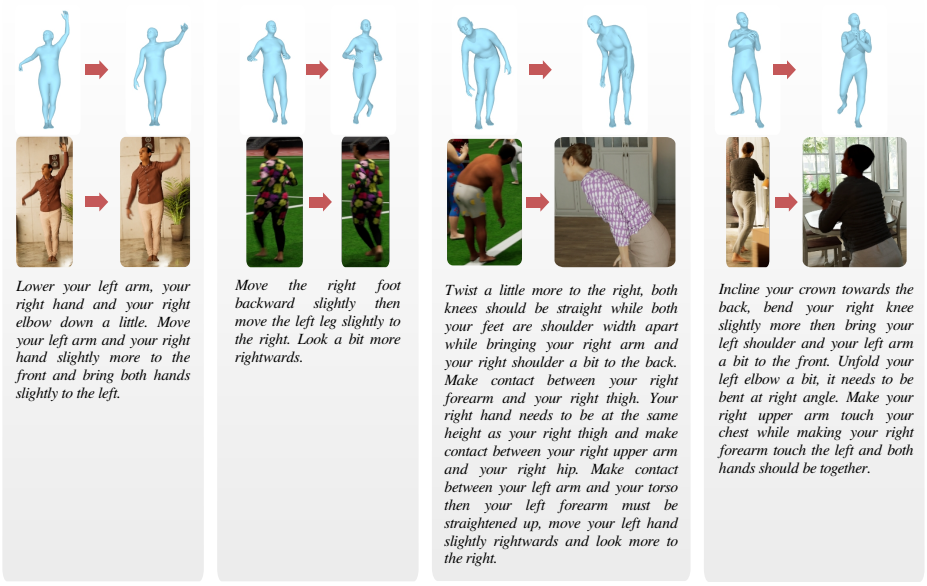

Fig. A2: Examples from BEDLAM-Fix. A textual instruction explains how to go from element A (left side of the arrow) to element B (right side of the arrow). Elements A and B can be either images and/or 3D poses.

rather on the right calf than the right thigh, but the confusion probably comes from the fact that the right wrist appears to be touching the right thigh). One recurrent behavior is the output of instructions like "move body part X to the right/left", which do not always seem to really apply, from a human perspective. This likely stems from the fact that the bodies in Figure A3 are only visible from the side – thus preventing a fair depth estimate.

Globally, the study of several qualitative examples reveals that the model struggles with lying down poses. For instance, in the middle example of the second row, it believes the ground is the left side of the image (hence "the left thigh and the right forearm need to be *parallel to the floor*"). This problem comes directly from the pose representation, and can be explained by the low number of lying down poses in the training data. The frequency of such poses in the training batches could be increased to mitigate this issue.

Other typical failure cases include when the person is only partially visible (*e.g.* lower body truncated by the image boundaries). In these conditions, the text generation model would often output average instructions about the legs and forget about the main differences of the upper body. One way to alleviate this limitation could be to *also* train the model on truncated images input (instead of 3D poses only), and to consider instructions that only mention differences about the visible body parts. Two main aspects of our method makes this conceivable. First, the PoseEmbroider can similarly treat image and 3D pose input: it provides a modality-agnostic representation to the text decoder. Second, the model can be trained efficiently on synthetic data, using the automatic pipeline from [4],

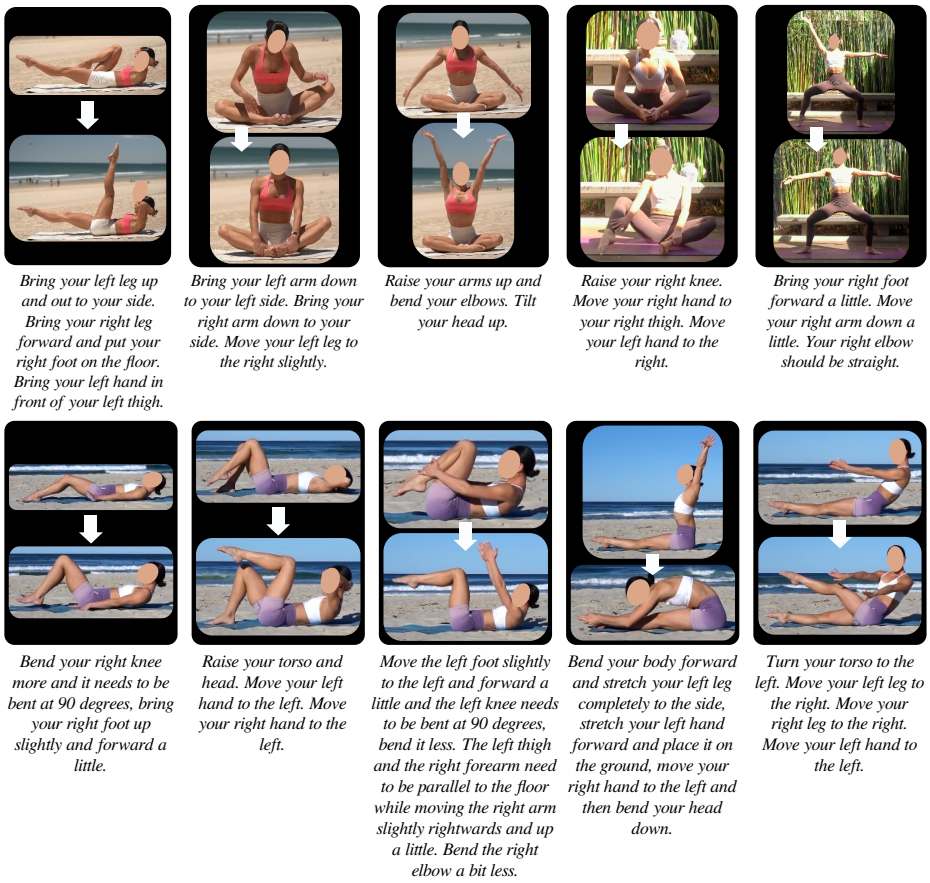

Fig. A3: Instruction generations on real-world images using the PoseEmbroider pose representation. The instruction generation model was trained using the PoseEmbroider representations of 3D poses only. The generated text is shown below each image pairs. We occluded the faces to preserve privacy.

which can be modified so as to produce instructions involving a specific set of body joints (*i.e.*, those visible in the images).

B.2 Any-to-any retrieval

We show a few more examples of any-to-any retrieval in Figure A4. We see that our model produces reasonable results for different types of query and target modalities.

C Original annotations for retrieval results

The original annotations for the queries presented in Figure 3 of the main paper were not displayed alongside the results; we provide them in Figure A5.

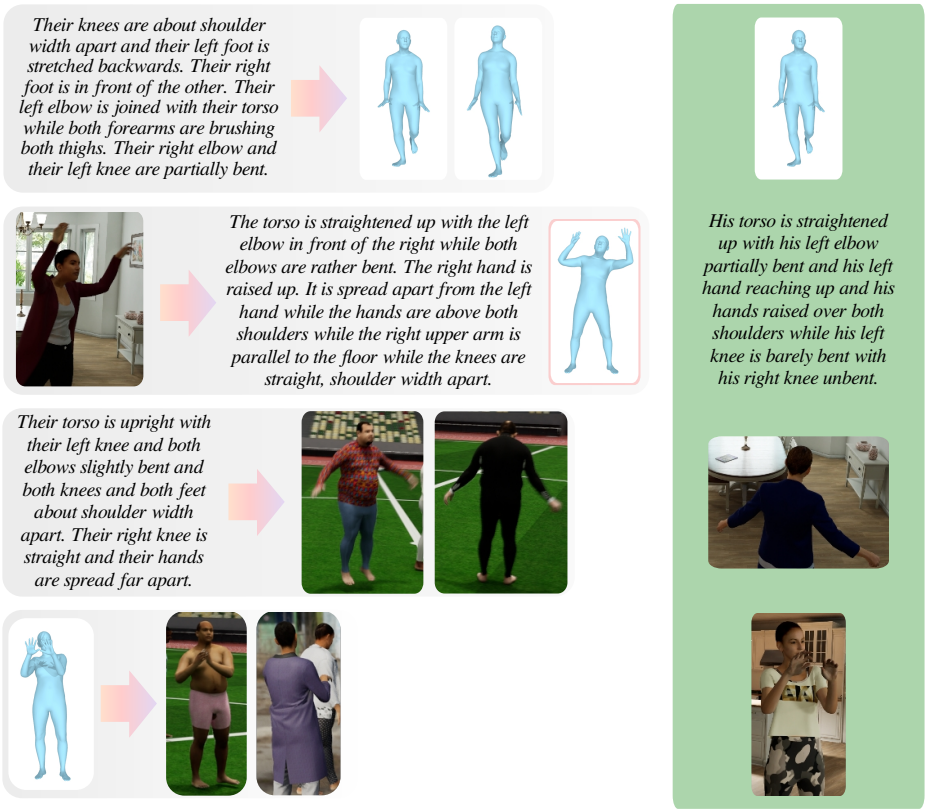

Fig. A4: Qualitative examples of any-to-any multi-modal retrieval on the validation split of BEDLAM-Script. We show either top-1 or top-2 results for several types of single input and output modalities. Original paired modalities for the queries on the left are shown in the green box on the right. To ease reading, we additionally show the 3D pose associated retrieved texts and give it a pink border.

D Implementation details

Architecture details. We detail below the architecture of each of our model components:

- *The pose encoder* is extracted from a Variational AutoEncoder (VAE) [10]. Its architecture follows VPoser [12], and was adapted to process the first 22 SMPL-X [12] body joint rotations (axis-angle representation). It shares the training objectives of the pose generative model in PoseFix [4], and has been trained on the 3D poses of BEDLAM-Script. For feature representation, we use the 512-dimensional vector that is further projected in the VAE to produce the distribution parameters. This frozen pretrained representation is fed to a trainable linear layer followed by a ReLU activation.
- *The text encoder* is the same as in [3]: text tokens are embedded thanks to a frozen DistilBERT [13], then fed to a transformer [14] (latent dimension

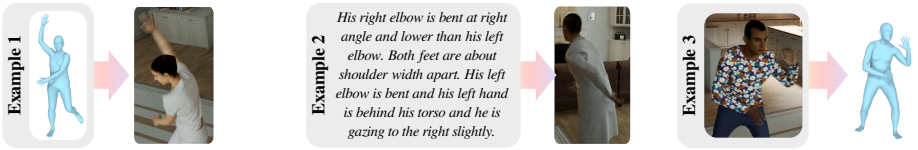

Fig. A5: Original paired modalities for the queries presented in Figure 3 of the main paper (qualitative any-to-any retrieval results).

512, 4 heads, 4 layers, feed-forward networks of size 1024, GELU [8] activations, dropout rate of 0.1). The final single-vector embedding of a text is obtained by average-pooling all its token encodings. This frozen pretrained representation is further given to a trainable linear layer followed by a ReLU activation.

- *The image encoder* is the Vision Transformer [5] backbone from the SMPLer-X [12] base model, which was connected with a neural head and trained end-to-end for human mesh recovery. This image encoder is thus assumed to yield already powerful human-aware visual features and kept frozen. While SMPLer-X reasons on all image patches at once for SMPL regression, we average pool the visual tokens during the pretraining of the PoseEmbroider, based solely on synthetic data, and do not tune them in later stages (*e.g.* for SMPL regression, in Section 6 of the main paper). Specifically, the frozen, pretrained patch representations are aggregated into a single-vector representation after going through a trainable linear layer (projecting into a 512-dimensional space), and a ReLU activation.
- *The Embroider model* is a transformer [14] (latent dimension 256, 4 heads, 4 layers, feed-forward networks of size 512, GELU [8] activations, dropout rate of 0.1) followed by a LayerNorm [1]. It is sandwiched by two linear layers, projecting the input from a 512-dimensional space to the 256-dimensional working space of the transformer and vice-versa. Learned tokens (*i.e.* x, e_v, e_p and e_t) are learnable parameters of size 512. Modality-specific re-projection MLPs, processing the PoseEmbroider output $\bar{x}$, consist in small multi-layer perceptrons [7] with 2 fully-connected layers of size 512 and a ReLU activation in-between. Their outputs are further L2-normalized.
- *The Aligner baseline model* appends modality-specific MLPs to each modality encoder. They consist of three linear layers with two in-between ReLU activations and dropout. Their hidden dimension is the same as the input, and they project into a 512-dimensional space.
- *The text decoder* of the text generation model has the same architecture as in PoseFix [4], except that it takes pose encodings of size 512 instead of 32. The pose encodings are fused with TIRG [15], projected thanks to a linear layer of dimension 512, then fed via cross-attentions to a transformer decoder (latent dimension 512, 8 heads, 4 layers, feed-forward networks of size 1024, GELU [8] activations, dropout rate of 0.1), which takes 512-dimensional token encodings as input. The output tokens are given to a linear layer of the size of the vocabulary to predict the likelihood of each subsequent word.

The PoseEmbroider and Aligner models have a similar size of 164.8M parameters (even though the reprojection heads of the PoseEmbroider are expandable), including 162.5M just for the encoders.

Optimization and training details. The PoseEmbroider model is trained for 350 epochs, with all the uni-modal encoders frozen. We use mini-batches of size 128, a learning rate of 2.10^{-4}, the Adam [9] optimizer and a learning rate scheduler considering steps of size 400 and a gamma value of 0.5.

The text generation model is optimized with Adam, with a learning rate and weight decay of 10^{-4}, for 900 epochs, and with batch sizes of 64. The finetuning on the PoseFix-OOS dataset and BEDLAM-Fix is run on 300 epochs. All trainings were done using precomputed cached features for the input pose representations.

R-precision metrics for text generation. This metric was originally proposed by [6] for motion-to-text generation, and directly imported by [4] for instruction text generation from pose pairs. It requires to first train an auxiliary retrieval model that links annotated texts and pose pairs. Then, for each generated text, this model is used to rank a set of pose pairs. The R-precision corresponds to the maximal rank of the pose pair that was actually used to generate the text. [4] followed [6] and used a pool size of 32, however we report results on a harder pool size of 200.

Text instruction generation model: comparison with PoseFix's [4]. We explain here the differences with the setting in [4], which prevents the direct comparison of the models presented in this paper with those from [4]. First, [4] trains the pose encoder from scratch alongside the text decoder, which results in a pose encoder finetuned for the studied task. In this work, as we aim to compare off-the-shelf representations and offer inference from image input, we resort to pretrained frozen (and thus potentially sub-optimal) pose encoders, however allowing a mapping to the visual space. Next, while we both use data derived from AMASS [11] (recall that BEDLAM [2] uses motions from AMASS), it is not exactly the same. In particular, the construction of BEDLAM-Script and BEDLAM-Fix had to account for selection criteria on the images as well (joint visibility, inter-person occlusions, crop resolution *etc.*). In addition, in this work we only consider 50k poses (and 54k pairs) for pretraining, against 100k poses (and 95k pairs) in [4].

Aside from the aforementioned elements, the Aligner in Table 2 of the main paper is the closest to PoseFix's text generation model, architecture-wise.

E Responsibility to human subjects

Our models were trained exclusively on synthetic data from BEDLAM [2] and data from PoseFix [4] which includes human-written texts, but those do not carry any personal information.

The real-world images used to showcase the capabilities of our text generation model are solely used for qualitative studies. The Yoga images from Figure 6 of the main paper were obtained in studio with the written agreement of the

subject. The images from Figure A3 were extracted from a public YouTube video, and we hide the faces to preserve anonymity.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 6
2. Black, M.J., Patel, P., Tesch, J., Yang, J.: BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In: CVPR (2023) 7
3. Delmas, G., Weinzaepfel, P., Lucas, T., Moreno-Noguer, F., Rogez, G.: PoseScript: 3D Human Poses from Natural Language. In: ECCV (2022) 5
4. Delmas, G., Weinzaepfel, P., Moreno-Noguer, F., Rogez, G.: PoseFix: Correcting 3D Human Poses with Natural Language. In: ICCV (2023) 3, 5, 6, 7
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 6
6. Guo, C., Zuo, X., Wang, S., Cheng, L.: Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In: ECCV (2022) 7
7. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall PTR (1994) 6
8. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016) 6
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015) 7
10. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014) 5
11. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: ICCV (2019) 7
12. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: CVPR (2019) 5, 6
13. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019) 5
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 5, 6
15. Vo, N., Jiang, L., Sun, C., Murphy, K., Li, L.J., Fei-Fei, L., Hays, J.: Composing text and image for image retrieval-an empirical odyssey. In: CVPR (2019) 6