PoseScript: 3D Human Poses from Natural Language

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

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

Abstract. Natural language is leveraged in many computer vision tasks such as image captioning, cross-modal retrieval or visual question answering, to provide fine-grained semantic information. While human pose is key to human understanding, current 3D human pose datasets lack detailed language descriptions. In this work, we introduce the PoseScript dataset, which pairs a few thousand 3D human poses from AMASS with rich human-annotated descriptions of the body parts and their spatial relationships. To increase the size of this dataset to a scale compatible with typical data hungry learning algorithms, we propose an elaborate captioning process that generates automatic synthetic descriptions in natural language from given 3D keypoints. This process extracts low-level pose information – the *posecodes* – using a set of simple but generic rules on the 3D keypoints. The posecodes are then combined into higher level textual descriptions using syntactic rules. Automatic annotations substantially increase the amount of available data, and make it possible to effectively pretrain deep models for finetuning on human captions. To demonstrate the potential of annotated poses, we show applications of the PoseScript dataset to retrieval of relevant poses from large-scale datasets and to synthetic pose generation, both based on a textual pose description. Code and dataset are available at https://europe.naverlabs.com/research/computer-vision/posescript/.

1 Introduction

'*The pose has the head down, ultimately touching the floor, with the weight of the body on the palms and the feet. The arms are stretched straight forward, shoulder width apart; the feet are a foot apart, the legs are straight, and the hips are raised as high as possible.*'. The text above describes the downward dog yoga pose[3], and a reader is able to picture such a pose from this natural language description. Being able to automatically map natural language descriptions and accurate 3D human poses would open the door to a number of applications such as helping image annotation when the deployment of Motion Capture (MoCap) systems is

[3] https://en.wikipedia.org/wiki/Downward_Dog_Pose

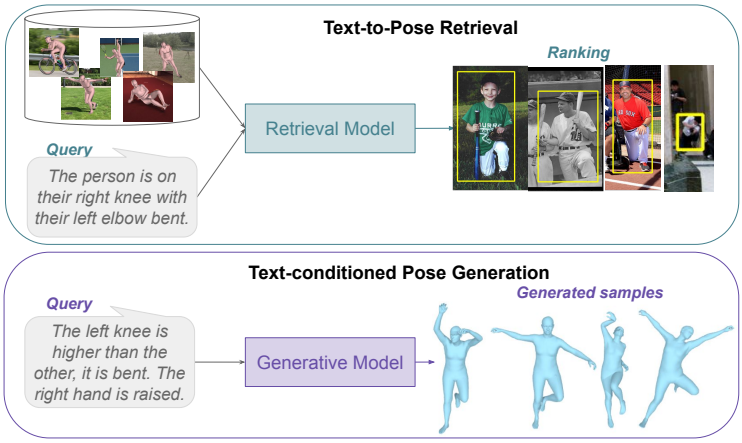


Fig. 1. Illustration of possible applications using PoseScript. The top figure illustrates text-to-pose retrieval where the goal is to retrieve poses in a large-scale database given a text query. This can be applied to databases of images with associated SMPL fits. The bottom figure shows an example of text-conditioned pose generation.

not practical; performing semantic searches in large-scale datasets (see Figure 1 top), which are currently only based on high-level metadata such as the action being performed [14,26,35]; complex pose or motion data generation in digital animation (see Figure 1 bottom); or teaching basic posture skills to visually impaired individuals [42].

While the problem of combining language and images or videos has attracted significant attention [17,43,21,10], in particular with the impressive results obtained by the recent multimodal neural networks CLIP [36] and DALL-E [37], the problem of linking text and 3D geometry is largely unexplored. There have been a few recent attempts at mapping text to rigid 3D shapes [8], and at using natural language for 3D object localization [7] or 3D object differentiation [1]. More recently, Fieraru *et al.* [11] introduce AIFit, an approach to automatically generate human-interpretable feedback on the difference between a reference and a target motion. There have also been a number of attempts to model humans using various forms of text. Attributes have been used for instance to model body shape [41] and face images [15]. Other approaches [12,2,31,3] leverage textual descriptions to generate motion, but without fine-grained control of the body limbs. More related to our work, Pavlakos *et al.* [29] exploit the relation between two joints along the depth dimension, and Pons-Moll *et al.* [34] describe 3D human poses through a series of *posebits*, which are binary indicators for different types of questions such as 'Is the right hand above the hips?'. However, these types of Boolean assertions have limited expressivity and remain far from the natural language descriptions a human would use.

In this paper, we propose to map 3D human poses with arbitrarily complex structural descriptions, in natural language, of the body parts and their spatial relationships. To that end, we first introduce the *PoseScript* dataset,

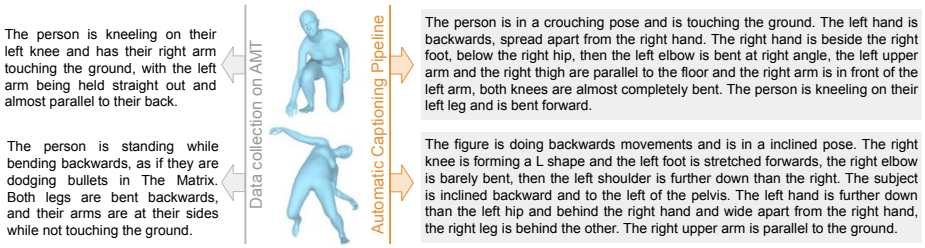


Fig. 2. Examples of pose descriptions from PoseScript, produced by human annotators (left) and by our automatic captioning pipeline (right).

which consists of captions written by human annotators for about 4,000 poses from the AMASS dataset [26]. To scale-up this dataset, we additionally propose an automatic captioning pipeline for human-centric poses that makes it possible to annotate thousands of human poses in a few minutes. Our pipeline is built on (a) low-level information obtained via an extension of posebits [34] to finer-grained categorical relations of the different body parts (*e.g.* 'the knees are slightly/relatively/completely bent'), units that we refer to as *posecodes*, and on (b) higher-level concepts that come either from the action labels annotated by the BABEL dataset [35], or combinations of posecodes. We define rules to select and aggregate posecodes using linguistic aggregation principles, and convert them into sentences to produce textual descriptions. As a result, we are able to automatically extract human-like captions for a normalized input 3D pose. Importantly, since the process is randomized, we can generate several descriptions per pose, as different human annotators would do. We used this procedure to describe 20,000 poses extracted from the AMASS dataset. Figure 2 shows examples of human-written and automatic captions.

Using the PoseScript dataset, we propose to tackle two tasks, see Figure 1. The first is a cross-modal retrieval task where the goal is to retrieve from a database the poses that are most similar to a given text query; this can also be applied to RGB images by associating them with 3D human fits. The second task consists in generating human poses conditioned on a textual description. In both cases, our experiments demonstrate that it is beneficial to pretrain models using the automatic captions before finetuning them on real captions.
In summary, our contributions are threefold:

- We introduce the PoseScript dataset (Section 3). It associates human poses and structural descriptions in natural language, either obtained through human-written annotations or using our automatic captioning pipeline.
- We then study the task of text-to-pose retrieval (Section 4).
- We finally present the task of text-conditioned pose generation (Section 5).

2 Related Work

Text for humans in images. Some previous works have used attributes as semantic-level representation to edit body shapes [41] or image faces [15]. In

contrast, our approach focuses on body poses and leverages natural language, which has the advantage of being unconstrained and more flexible. Closer to our work, [46,6] focus on generating human 2D poses, SMPL parameters or even images from captions. However, they use MS Coco [24] captions, which are generally simple image-level statements on the activity performed by the human, and which sometimes relate to the interaction with other elements from the scene, *e.g.* 'A soccer player is running while the ball is in the air'. In contrast, we focus on fine-grained detailed captions about the pose only. FixMyPose [18] provides manually annotated captions about the difference between human poses in two synthetic images. These captions also mention objects from the environment, *e.g.* 'carpet' or 'door'. Similarly, AIFit [11] proposes to automatically generate text about the discrepancies between a reference motion and a performed one, based on differences of angles and positions. We instead focus on describing one single pose without relying on any other visual element.

Text for human motion. We deal with static poses, whereas several existing methods have mainly studied 3D action (sequence) recognition or text-based 2D [2] or 3D motion synthesis. They either condition their model on action labels [13,31,25], or descriptions in natural language [33,45,23,3,12]. Yet, even if motion descriptions effectively constrain *sequences* of poses, they do not specifically inform about individual poses. What if an animation studio looks for a sequence of 3D body poses where 'the man is running with his hands on his hips'? The model used by the artists to initialize the animation should have a deep understanding of the relations between the body parts. To this end, it is important to learn about specific pose semantics, beyond global pose sequence semantics.

Pose semantic representations. Our captioning generation process relies on posecodes that capture relevant information about the pose semantics. Posecodes are inspired from posebits [34] where images showing a human are annotated with various binary indicators. This data is used to reduce ambiguities in 3D pose estimation. Conversely, we automatically extract posecodes from normalized 3D poses in order to generate descriptions in natural language. Ordinal depth [29] can be seen as a special case of posebits, focusing on the depth relationship between two joints. They obtain annotations on some training images to improve a human mesh recovery model by adding extra constraints. Poselets [5] can also be seen as another way to extract discriminative pose information, but are not easily interpreted. In contrast to these representations, we propose to generate pose descriptions in natural language, which have the advantage (a) of being a very intuitive way to communicate ideas, and (b) of providing greater flexibility.

In summary, our proposed PoseScript dataset differs from existing datasets in that it focuses on single 3D poses instead of motion [32], and provides direct descriptions in natural language instead of simple action labels [35,13,40,22,14], binary relations [34,29] or modifying texts [18,11]. To the best of our knowledge, this is the first attempt at associating static 3D poses and descriptions in natural language.

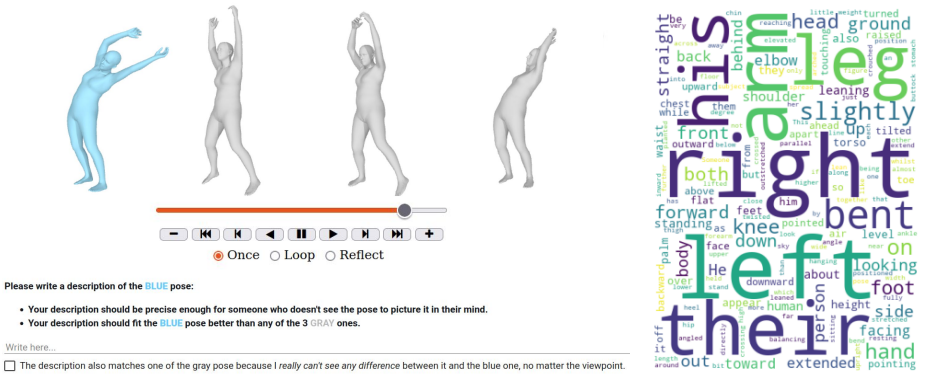


Fig. 3. Left: Interface presented to the AMT annotators in order to collect discriminative descriptions of the blue pose. **Right: Wordcloud** of the most frequent words in the human-written descriptions.

3 The PoseScript Dataset

The PoseScript dataset is composed of static 3D human poses, together with fine-grained semantic annotations in natural language. We provide **H**uman-written annotated descriptions (PoseScript-H), and further increase the amount of data with **A**utomatically generated captions (PoseScript-A). The crowd-sourced data collection process is described in Section 3.1, and the automatic captioning pipeline in Section 3.2. Finally, aggregated statistics over the PoseScript dataset are provided in Section 3.3.

3.1 Dataset collection

We collect human-written captions for 3D human poses extracted from the AMASS dataset [26], using Amazon Mechanical Turk[4] (AMT), a crowd-sourced annotation platform. The interface, displayed in Figure 3 (left), presents the annotators with the mesh of the human pose to annotate (in blue), and a slider to control the viewpoint. To encourage discriminative captions, we additionally display 3 discriminator poses (in gray), which are semantically close to the pose to annotate. The task is to provide a description of the blue pose which is precise enough to distinguish it from the three others. We detail the discriminator selection, the complete task instructions and annotator information in the supplementary material. Some PoseScript-H examples are shown in Figure 2 (left).

3.2 Automatic captioning pipeline

We now describe the process used to generate synthetic textual descriptions for 3D human poses. As depicted in Figure 4, it relies on the extraction, selection and aggregation of elementary pieces of pose information, called *posecodes*, that are eventually converted into sentences to produce a description.

[4] https://www.mturk.com

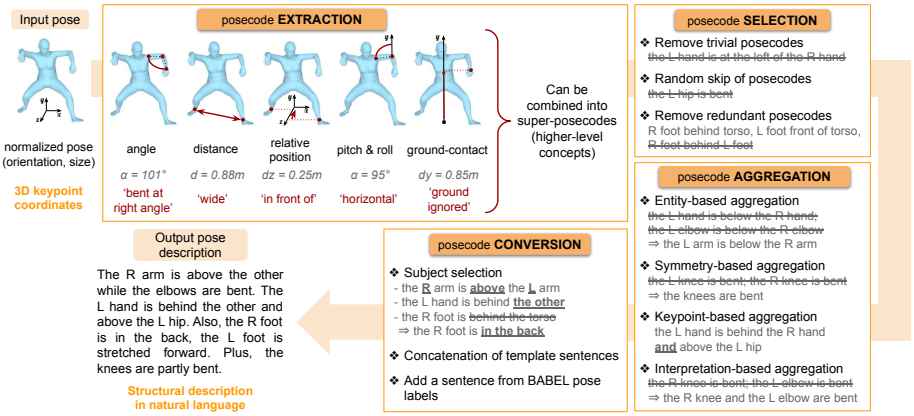


Fig. 4. Overview of our captioning pipeline. Given a normalized 3D pose, we use posecodes to extract semantic pose information. These posecodes are then selected, merged or combined (when relevant) before being converted into a structural pose description in natural language. Letters 'L' and 'R' stand for 'left' and 'right' respectively.

The process takes 3D keypoint coordinates of human-centric poses as input. These are inferred with the SMPL-H body model [38] using the default shape coefficients and a normalized global orientation along the y-axis.

1. Posecode extraction. A posecode describes a relation between a specific set of joints. We capture five kinds of elementary relations: angles, distances and relative positions (as in [34]), but also pitch, roll and ground-contacts.

◦ *Angle posecodes* describe how a body part 'bends' at a given joint, *e.g.* the left elbow. Depending on the angle, the posecode is assigned one of the following attributes: '`straight`', '`slightly bent`', '`partially bent`', '`bent at right angle`', '`almost completely bent`' and '`completely bent`'.

◦ *Distance posecodes* categorize the $L2$-distance between two keypoints (*e.g.* the two hands) into '`close`', '`shoulder width apart`', '`spread`' or '`wide`' apart.

◦ *Posecodes on relative position* compute the difference between two keypoints along a given axis. The possible categories are, for the x-axis: '`at the right of`', '`x-ignored`', '`at the left of`'; for the y-axis: '`below`', '`y-ignored`', '`above`'; and for the z-axis: '`behind`', '`z-ignored`' and '`in front of`'. In particular, comparing the x-coordinate of the left and right hands allows to infer if they are crossed (*i.e.*, the left hand is '`at the right`' of the right hand). The '`ignored`' interpretations are ambiguous configurations which will not be described.

◦ *Pitch & roll posecodes* assess the verticality or horizontality of a body part defined by two keypoints (*e.g.* the left knee and hip together define the left thigh). A body part is '`vertical`' if it is approximately orthogonal to the y-hyperplane, and '`horizontal`' if it is in it. Other configurations are '`pitch-roll-ignored`'.

◦ *Ground-contact posecodes*, used for intermediate computation only, denote whether a keypoint is '`on the ground`' (*i.e.*, vertically close to the keypoint of minimal height in the body, considered as the ground) or '`ground-ignored`'.

Handling ambiguity in posecode categorization. Posecode categorizations are obtained using predefined thresholds. As these values are inherently subjective, we randomize the binning step by also defining a noise level applied to the measured angles and distances values before thresholding.

Higher-level concepts. We additionally define a few *super-posecodes* to extract higher-level pose concepts. These posecodes are binary (they either apply or not to a given pose configuration), and are expressed from elementary posecodes. For instance, the super-posecode '`kneeling`' can be defined as having both knees '`on the ground`' and '`completely bent`'.

2. Posecode selection aims at selecting an interesting subset of posecodes among those extracted, to obtain a concise yet discriminative description. First, we remove trivial settings (*e.g.* 'the left hand is at the left of the right hand'). Next, based on a statistical study over the whole set of poses, we randomly skip a few non-essential –*i.e.*, non-trivial but non highly discriminative – posecodes, to account for natural human oversights. We also set highly-discriminative posecodes as unskippable. Finally, we remove redundant posecodes based on statistically frequent pairs and triplets of posecodes, and transitive relations between body parts. Details are provided in the supplementary material.

3. Posecode aggregation consists in merging together posecodes that share semantic information. This reduces the size of the caption and makes it more natural. We propose four specific aggregation rules:

◦ *Entity-based aggregation* merges posecodes that have similar categorizations while describing keypoints that belong to a larger entity (*e.g.* the arm or the leg). For instance 'the left hand is below the right hand' + 'the left elbow is below the right hand' is combined into 'the left arm is below the right hand'.

◦ *Symmetry-based aggregation* fuses posecodes that share the same categorization, and operate on joint sets that differ only by their side of the body. The joint of interest is hence put in plural form, *e.g.* 'the left elbow is bent' + 'the right elbow is bent' becomes 'the elbows are bent'.

◦ *Keypoint-based aggregation* brings together posecodes with a common keypoint. We factor the shared keypoint as the subject and concatenate the descriptions. The subject can be referred to again using *e.g.* 'it' or 'they'. For instance, 'the left elbow is above the right elbow' + 'the left elbow is close to the right shoulder' + 'the left elbow is bent' is aggregated into 'The left elbow is above the right elbow, and close to the right shoulder. It is bent.'.

◦ *Interpretation-based aggregation* merges posecodes that have the same categorization, but apply on different joint sets (that may overlap). Conversely to entity-based aggregation, it does not require that the involved keypoints belong to a shared entity. For instance, 'the left knee is bent' + 'right elbow is bent' becomes 'the left knee and the right elbow are bent'.

Aggregation rules are applied at random when their conditions are met. In particular, joint-based and interpretation-based aggregation rules may operate on the same posecodes. To avoid favouring one rule over the other, merging options are first listed together and then applied at random.

4. Posecode conversion into sentences is performed in two steps. First, we select the subject of each posecode. For symmetrical posecodes – which involve two joints that only differ by their body side – the subject is chosen at random between the two keypoints, and the other is randomly referred to by its name, its side or 'the other' to avoid repetitions and provide more varied captions. For asymmetrical posecodes, we define a 'main' keypoint (chosen as subject) and 'support' keypoints, used to specify pose information (*e.g.* the 'head' in 'the left hand is raised above the head'). For the sake of flow, in some predefined cases, we omit to name the support keypoint (*e.g.* 'the left hand is raised above the head' is reduced to 'the left hand is raised'). Second, we combine all posecodes together in a final aggregation step. We obtain individual descriptions by plugging each posecode information into one template sentence, picked at random in the set of possible templates for a given posecode category. Finally, we concatenate the pieces in random order, using random pre-defined transitions. Optionally, for poses extracted from annotated sequences in BABEL [35], we add a sentence based on the associated high-level concepts (*e.g.* 'the person is in a yoga pose').

Some automatic captioning examples are presented in Figure 2 (right). The captioning process is highly modular; it allows to simply define, select and aggregate the posecodes based on different rules. Design of new kinds of posecodes (especially super-posecodes) or additional aggregation rules, can yield further improvements in the future. Importantly, randomization has been included at each step of the pipeline which makes it possible to generate different captions for the same pose, as a form of data augmentation, see supplementary material.

3.3 Dataset statistics

The PoseScript dataset contains a total of 20,000 human poses sampled from the AMASS dataset using a farthest-point sampling algorithm to maximize the variability. Specifically, we first infer the joint positions for each pose in a normalized way, using the neutral body model with the default shape coefficients and the global orientation set to 0. Then, starting from one random pose in the dataset, we iteratively select the pose with the maximum MPJE (mean per-joint error) to the set of poses that were already selected.

We collected 3.893 human annotations on AMT (PoseScript-H). We semi-automatically clean the descriptions by manually correcting the spelling of words that are not in the English dictionary, by removing one of two identical consecutive words, and by checking the error detected by a spell checker, namely NeuSpell [27]. Human-written descriptions have an average length of 55.1 tokens (51.4 words, plus punctuation). An overview of the most frequent words, among a vocabulary of 1654, is presented in Figure 3 (right).

We used the automatic captioning pipeline to increase the number of pose descriptions in the dataset (PoseScript-A). We designed a total of 87 posecodes, and automatically generated 6 captions for each of the 20,000 poses, in less than 6 minutes. Overall, automatic descriptions were produced using a posecode skipping rate of 15%, and an aggregation probability of 95%. Further details about the posecodes and other dataset statistics are provided in the supplementary.

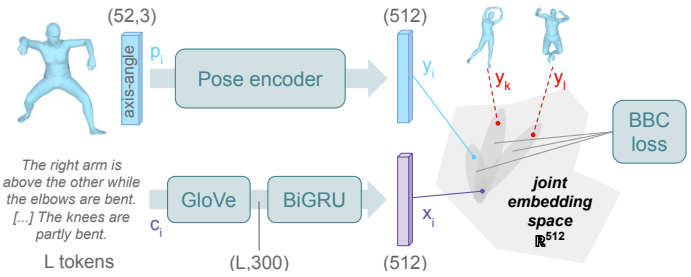


Fig. 5. Overview of the training scheme of the retrieval model. The input pose and caption are fed to a pose encoder and a text encoder respectively to map them into a joint embedding space. The loss encourages the pose embedding y_i and its caption embedding x_i to be close in this latent space, while being pulled apart from features of other poses in the same training batch (*e.g.* y_k and y_l).

We split the dataset into roughly 70% for training, 10% for validation and 20% for testing while ensuring that poses from the same AMASS sequence belong to the same split. When considering the automatic captions, we obtain 14,004 poses for training, 2,025 for validation and 3,971 for testing. When considering the human-written captions, each split respectively includes 2,713 (train), 400 (validation) and 780 (test) human-annotated poses.

4 Application to Text-to-Pose Retrieval

In this section, we study the problem of *text-to-pose retrieval*, which consists in ranking a large collection of poses by relevance to a given textual query (and likewise for pose-to-text retrieval). In such cross-modal retrieval task, it is standard to encode the multiple modalities into a common latent space.

Problem formulation. Let $S = \{(c_i, p_i)\}_{i=1}^{N}$ be a set of caption-and-pose pairs. By construction, p_i is the most relevant pose for caption c_i, which means that $p_{j\neq i}$ should be ranked after p_i for text-to-pose retrieval. In other words, the retrieval model aims to learn a similarity function $s(c, p) \in \mathbb{R}$ such that $s(c_i, p_i) > s(c_i, p_{j\neq i})$. As a result, a set of relevant poses can be retrieved for a given text query by computing and ranking the similarity scores between the query and each pose from the collection (the same goes for pose-to-text retrieval).

Since poses and captions are from two different modalities, we first use modality-specific encoders to embed the inputs into a joint embedding space, where the two representations will be compared to produce the similarity score.

Let $\theta(\cdot)$ and $\phi(\cdot)$ be the textual and pose encoders respectively. We denote as $x = \theta(c) \in \mathbb{R}^d$ and $y = \phi(p) \in \mathbb{R}^d$ the $L2$-normalized representations of a caption c and of a pose p in the joint embedding space (see Figure 5).

Encoders. The tokenized caption is embedded by a bi-GRU [9] taking pre-trained GloVe word embeddings [30] as input. The pose is first encoded as a matrix of size $(52, 3)$, consisting in the rotation of the SMPL-H body joints in

	mRecall↑	pose-to-text			text-to-pose		
		R@1↑	*R*@5↑	*R*@10↑	*R*@1↑	*R*@5↑	*R*@10↑
test on PoseScript-A (3,971 samples)							
trained on PoseScript-A	**72.4**$_{\pm 1.7}$	**44.8**$_{\pm 2.2}$	**75.9**$_{\pm 1.8}$	**85.2**$_{\pm 1.3}$	**54.4**$_{\pm 2.6}$	**83.6**$_{\pm 1.5}$	**90.4**$_{\pm 0.9}$
test on PoseScript-H (780 samples)							
trained on PoseScript-A	7.7$_{\pm 0.8}$	3.2$_{\pm 0.4}$	9.5$_{\pm 1.3}$	14.0$_{\pm 1.2}$	1.8$_{\pm 0.4}$	6.4$_{\pm 1.0}$	11.2$_{\pm 0.7}$
trained on PoseScript-H	13.1$_{\pm 0.4}$	4.4$_{\pm 0.4}$	14.6$_{\pm 0.4}$	21.3$_{\pm 0.7}$	3.7$_{\pm 0.3}$	13.1$_{\pm 1.0}$	21.6$_{\pm 0.7}$
trained on PoseScript-A, FT on PoseScript-H	**31.0**$_{\pm 1.2}$	**12.4**$_{\pm 0.7}$	**32.7**$_{\pm 1.3}$	**44.2**$_{\pm 1.6}$	**13.8**$_{\pm 1.1}$	**35.3**$_{\pm 1.6}$	**47.4**$_{\pm 1.5}$

Table 1. Text-to-pose and pose-to-text retrieval results on the test split of the PoseScript dataset. For human-written captions (PoseScript-H), we evaluate models trained on each specific caption set alone, and one pretrained on automatic captions (PoseScript-A) then finetuned (FT) on human captions. Results are averaged over 3 runs.

axis-angle representation. The pose is then flattened and fed as input to the pose encoder, chosen as the VPoser encoder [28]: it consists of a 2-layer MLP with 512 units, batch normalization and leaky-ReLU, followed by a fully-connected layer of 32 units. We add a ReLU and a final projection layer to produce an embedding of the same size d as the text encoding.

Training. Given a batch of B training pairs (x_i, y_i), we use the Batch-Based Classification (BBC) loss which is common in cross-modal retrieval [44]:

$$\mathcal{L}_{\text{BBC}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\big(\gamma\sigma(x_i, y_i)\big)}{\sum_j \exp\big(\gamma\sigma(x_i, y_j)\big)}, \tag{1}$$

where γ is a learnable temperature parameter and σ is the cosine similarity function $\sigma(x, y) = x^\top y / \big(\|x\|_2 \times \|y\|_2\big)$.

Evaluation protocol. Text-to-pose retrieval is evaluated by ranking the whole set of poses for each of the query texts. We then compute the recall@K ($R@K$), which is the proportion of query texts for which the corresponding pose is ranked in the top-K retrieved poses. We proceed similarly to evaluate pose-to-text retrieval. We use K = 1, 5, 10 and additionally report the mean recall (mRecall) as the average over all recall@K values from both retrieval directions.

Quantitative results. We report results on the test set of PoseScript in Table 1, both on automatic and human-written captions. Our model trained on automatic captions obtains a mean recall of 72.4%, with a R@1 above 40% and a R@10 above 80% on automatic captions. However, the performance degrades on human captions, as many words from the richer human vocabulary are unseen during training on automatic captions. When trained on human captions, the model obtains a higher – but still rather low – performance. Using human captions to finetune the initial model trained on automatic ones brings an improvement of a factor 2 and more, with a mean recall (resp. R@10 for text-to-pose) of 31.0% (resp. 47.4%) compared to 13.1% (resp. 21.6%) when training from scratch. This experiment clearly shows the benefit of using the automatic captioning pipeline to scale-up the PoseScript dataset. In particular, this suggests that the

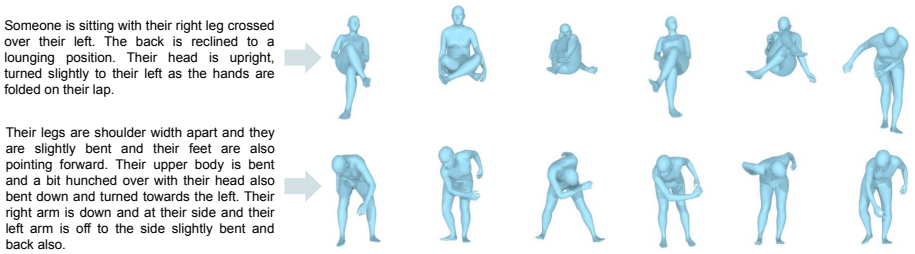


Fig. 6. Text-to-pose retrieval results for human-written captions from the PoseScript dataset. Directions such as 'left' and 'right' are relative to the body.

model is able to derive new concepts in human-written captions from non-trivial combination of existing posecodes in automatic captions.

Qualitative retrieval results. Examples of text-to-pose retrieval results are presented in Figure 6. It appears that the model is able to encode several pose concepts concurrently and to distinguish between the left and right body parts.

Retrieval in image databases. MS Coco [24] is one of several real-world datasets that have been used for human mesh recovery. We resort to the 74,834 pseudo-ground-truth SMPL fits provided by EFT [16], on which we apply our text-to-pose retrieval model trained with PoseScript. We then retrieve 3D poses among this MS Coco-EFT set, and display the corresponding images with the associated bounding box around the human body. Results are shown in Figure 7. We observe that overall, the constraints specified in the query text are satisfied in the images. Retrieval is based on the poses and not on the context, hence the third image of the first row where the pose is close to an actual kneeling one. This shows one application of a retrieval model trained on the PoseScript dataset: specific pose retrieval in images. Our model can be applied to any dataset of images containing humans, as long as SMPL fits are also available.

5 Application to Text-Conditioned Pose Generation

We next study the problem of *text-conditioned human pose generation*, *i.e.*, generating possible matching poses for a given text query. Our proposed model is based on Variational Auto-Encoders (VAEs) [20].

Training. Our goal is to generate a pose $\hat{p}$ given its caption c. To this end, we train a conditional VAE model that takes a tuple (p, c) composed of a pose p and its caption c at training time. Figure 8 gives an overview of our model. A pose encoder maps the pose p to a posterior over latent variables by producing the mean $\mu(p)$ and variance $\Sigma(p)$ of a normal distribution $\mathcal{N}_p = \mathcal{N}(\cdot|\mu(p), \Sigma(p))$. Another encoder is used to obtain a prior distribution $\mathcal{N}_c$, independent of p but conditioned on c. A latent variable $z \sim \mathcal{N}_p$ is sampled from $\mathcal{N}_p$ and decoded into a reconstructed pose $\hat{p}$. The training loss combines a reconstruction term $\mathcal{L}_R(p, \hat{p})$ between the original and reconstructed poses, p and $\hat{p}$ and a regularization term,

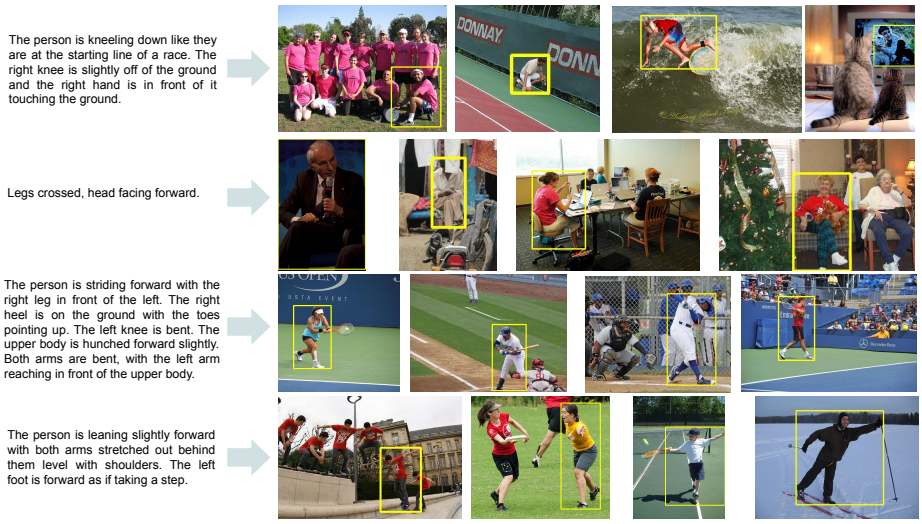


Fig. 7. Retrieval results in image databases. We use our text-to-pose retrieval model trained on human captions from PoseScript to retrieve 3D poses from SMPL fits on MS Coco, for some given text queries. We display the corresponding pictures for the top retrieved poses, along with the bounding boxes around the pose.

the Kullback-Leibler (KL) divergence between $\mathcal{N}_p$ and the prior $\mathcal{N}_c$:

$$\mathcal{L} = \mathcal{L}_R(p, \hat{p}) + \mathcal{L}_{KL}(\mathcal{N}_p, \mathcal{N}_c). \tag{2}$$

We also experiment with an additional loss term, $\mathcal{L}_{KL}(\mathcal{N}_p, \mathcal{N}(\cdot|0, I))$ which is a KL divergence between the posterior and the standard Gaussian $\mathcal{N}_0 = \mathcal{N}(\cdot|0, I)$. It can be seen as another regularizer and it also allows to sample poses from the model without conditioning on captions. We treat the variance of the decoder as a learned constant [39] and use a negative log likelihood (nll) as reconstruction loss, either from a Gaussian – which corresponds to an L2 loss and a learned variance term – or a Laplacian density, which corresponds to an L1 loss. Following VPoser, we use SMPL(-H) inputs with the axis-angle representation, and output joint rotations with the continuous 6D representation of [47]. Our reconstruction loss $\mathcal{L}_R(p, \hat{p})$ is a sum of the reconstruction losses between the rotation matrices – evaluated with a Gaussian log-likelihood – the position of the joints and the position of the vertices, both evaluated with a Laplacian log-likelihood.

Text-conditioned generation. At test time, a caption c is encoded into $\mathcal{N}_c$, from which z is sampled and decoded into a generated pose $\hat{p}$.

Evaluation metrics. We evaluate sample quality following the principle of the Fréchet inception distance: we compare the distributions of features extracted using our retrieval model (see Section 4), using real test poses and poses generated from test captions. This is denoted FID with an abuse of notation. We also report the mean-recall of retrieval models trained on real poses and evaluated on generated poses (mR R/G), and vice-versa (mR G/R). Both metrics

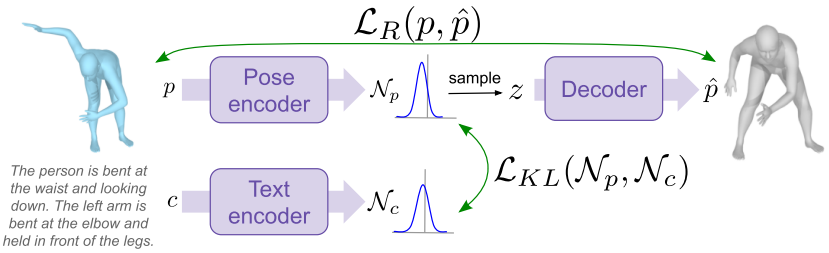


Fig. 8. Overview of the text-conditioned generative model. During training, it follows a VAE but where the latent distribution $\mathcal{N}_p$ from the pose encoder has a KL divergence term with the prior distribution $\mathcal{N}_c$ given by the text encoder. At test time, the sample z is drawn from the distribution $\mathcal{N}_c$.

	FID↓	ELBO jts↑	ELBO vert.↑	ELBO rot.↑	mRecall R/G↑	mRecall G/R↑
evaluation on automatic captions (PoseScript-A)						
without $\mathcal{L}_{KL}(\mathcal{N}_p, \mathcal{N}_0)$	**0.46**$\pm$0.00	**1.06**$\pm$0.00	**1.36**$\pm$0.01	**0.74**$\pm$0.00	28.73$\pm$1.99	46.83$\pm$5.41
with $\mathcal{L}_{KL}(\mathcal{N}_p, \mathcal{N}_0)$	0.48$\pm$0.01	1.05$\pm$0.00	1.35$\pm$0.00	**0.74**$\pm$0.01	**29.37**$\pm$1.84	**48.97**$\pm$4.39
evaluation on human captions (PoseScript-H) for the model with $\mathcal{L}_{KL}(\mathcal{N}_p, \mathcal{N}_0)$						
without pretraining	0.53$\pm$0.00	0.21$\pm$0.02	0.92$\pm$0.01	0.40$\pm$0.00	5.70$\pm$2.05	11.93$\pm$0.69
with pretraining	**0.48**$\pm$0.01	**0.47**$\pm$0.05	**1.11**$\pm$0.00	**0.47**$\pm$0.02	**18.23**$\pm$1.72	**28.27**$\pm$1.53

Table 2. Evaluation of the text-conditioned generative model on PoseScript-A for a model without or with $\mathcal{L}_{KL}(\mathcal{N}_p, \mathcal{N}_0)$ (top) and on PoseScript-H without or with pretraining on PoseScript-A (bottom). For comparison, the mRecall when training and testing on real poses is 72.4 with PoseScript-A and 31.0 on PoseScript-H. Results are averaged over 3 runs. The variability of R/G (resp. G/R) mRecall is due to the random selection of a generated pose sample at test (resp. training) time.

are sensitive to sample quality: the retrieval model will fail if the data is unrealistic. The second metric is also sensitive to diversity: missing parts of the data distribution hinder the retrieval model trained on samples. Finally, we report the Evidence Lower Bound (ELBO) computed on joints, vertices or rotation matrices, normalized by the target dimension.

Results. We present quantitative results in Table 2. We first find that adding the extra-regularization loss $\mathcal{L}_{KL}(\mathcal{N}_p, \mathcal{N}_0)$ to the model trained and evaluated on automatic captions has a low impact. Since it is convenient to sample poses without any conditioning, we keep this configuration and evaluate it when (a) training on human captions and (b) pretraining on automatic captions and finetuning on human captions. Pretraining improves all metrics, in particular retrieval testing and ELBOs improve substantially: pretraining helps to yield realistic and diverse samples. We display generated samples in Figure 9; the poses are realistic and generally correspond to the query. There are some variations, especially when the caption allows it, for instance with the position of the left arm in the top example or the height of the right leg in the third row. Failure cases can happen;

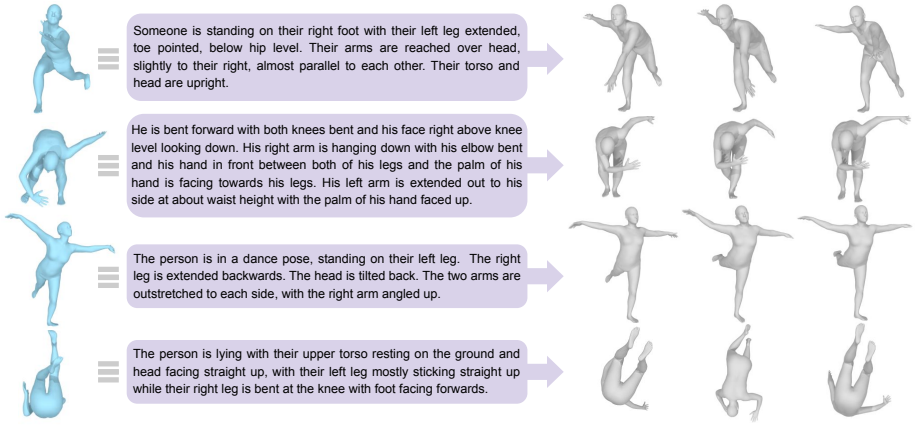


Fig. 9. Examples of generated samples. We show several generated samples (in grey) obtained for the human-written captions presented in the middle. For reference, we also show in blue the pose for which this annotation was originally collected.

Fig. 10. Example of potential application to SMPL fitting in images. Using the text-conditional pose prior (right) yields a more accurate 3D pose than a generic pose prior (left) when running the optimization-based SMPL fitting method SMPLify.

in particular rare words like 'lying' in the bottom row lead to higher variance in the generated samples; some of them are nevertheless close to the reference.

Application to SMPL fitting in image. We showcase the potential of leveraging text data for 3D tasks on a challenging example from SMPLify [4], in Figure 10. We use our text-conditional prior instead of the generic VPoser prior [28] to initialize to a pose closer to the ground truth and to better guide the in-the-loop optimization, which helps to avoid bad local minima traps.

6 Conclusion

We introduced PoseScript, the first dataset to map 3D human poses and structural descriptions in natural language. We provided applications to text-to-pose retrieval and to text-conditioned human pose generation. For both tasks, performance is improved by pretraining on the automatic captions. Future avenues on this topic include generating images from the generated poses or exploring motion generation conditioned on complex textual description.

Acknowledgements. This work is supported in part by the Spanish government with the project MoHuCo PID2020-120049RB-I00, and by Naver Labs Europe under the technology transfer contract 'Text4Pose'.

References

1. Achlioptas, P., Fan, J., Hawkins, R., Goodman, N., Guibas, L.J.: Shapeglot: Learning language for shape differentiation. In: ICCV (2019)
2. Ahn, H., Ha, T., Choi, Y., Yoo, H., Oh, S.: Text2action: Generative adversarial synthesis from language to action. In: ICRA (2018)
3. Ahuja, C., Morency, L.P.: Language2pose: Natural language grounded pose forecasting. 3DV (2019)
4. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In: ECCV (2016)
5. Bourdev, L., Malik, J.: Poselets: Body part detectors trained using 3D human pose annotations. In: ICCV (2009)
6. Briq, R., Kochar, P., Gall, J.: Towards better adversarial synthesis of human images from text. arXiv preprint arXiv:2107.01869 (2021)
7. Chen, D.Z., Chang, A.X., Nießner, M.: Scanrefer: 3D object localization in rgb-d scans using natural language. In: ECCV (2020)
8. Chen, K., Choy, C.B., Savva, M., Chang, A.X., Funkhouser, T., Savarese, S.: Text2shape: Generating shapes from natural language by learning joint embeddings. In: ACCV (2018)
9. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: EMNLP (2014)
10. Feng, F., Wang, X., Li, R.: Cross-modal retrieval with correspondence autoencoder. In: ACMMM (2014)
11. Fieraru, M., Zanfir, M., Pirlea, S.C., Olaru, V., Sminchisescu, C.: AIFit: Automatic 3D human-interpretable feedback models for fitness training. In: CVPR (2021)
12. Ghosh, A., Cheema, N., Oguz, C., Theobalt, C., Slusallek, P.: Synthesis of compositional animations from textual descriptions. In: ICCV (2021)
13. Guo, C., Zuo, X., Wang, S., Zou, S., Sun, Q., Deng, A., Gong, M., Cheng, L.: Action2motion: Conditioned generation of 3D human motions. In: ACMMM (2020)
14. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans. PAMI (2014)
15. Jiang, Y., Huang, Z., Pan, X., Loy, C.C., Liu, Z.: Talk-to-edit: Fine-grained facial editing via dialog. In: ICCV (2021)
16. Joo, H., Neverova, N., Vedaldi, A.: Exemplar fine-tuning for 3D human model fitting towards in-the-wild 3D human pose estimation. In: 3DV (2020)
17. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
18. Kim, H., Zala, A., Burri, G., Bansal, M.: FixMyPose: Pose correctional captioning and retrieval. In: AAAI (2021)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014)
21. Li, S., Xiao, T., Li, H., Yang, W., Wang, X.: Identity-aware textual-visual matching with latent co-attention. In: ICCV (2017)
22. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: CVPR Workshops (2010)

23. Lin, A.S., Wu, L., Corona, R., Tai, K.W.H., Huang, Q., Mooney, R.J.: Generating animated videos of human activities from natural language descriptions. In: NeurIPS workshops (2018)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
25. Lucas, T., Baradel, F., Weinzaepfel, P., Rogez, G.: PoseGPT: Quantizing human motion for large scale generative modeling. In: ECCV (2022)
26. Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., Black, M.J.: AMASS: Archive of motion capture as surface shapes. In: ICCV (2019)
27. Muralidhar Jayanthi, S., Pruthi, D., Neubig, G.: Neuspell: A neural spelling correction toolkit. In: EMNLP (2020)
28. 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)
29. Pavlakos, G., Zhou, X., Daniilidis, K.: Ordinal depth supervision for 3D human pose estimation. In: CVPR (2018)
30. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP (2014)
31. Petrovich, M., Black, M.J., Varol, G.: Action-conditioned 3D human motion synthesis with transformer VAE. In: ICCV (2021)
32. Plappert, M., Mandery, C., Asfour, T.: The kit motion-language dataset. Big data (2016)
33. Plappert, M., Mandery, C., Asfour, T.: Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. Robotics Auton. Syst. (2018)
34. Pons-Moll, G., Fleet, D.J., Rosenhahn, B.: Posebits for monocular human pose estimation. In: CVPR (2014)
35. Punnakkal, A.R., Chandrasekaran, A., Athanasiou, N., Quiros-Ramirez, A., Black, M.J.: BABEL: Bodies, action and behavior with english labels. In: CVPR (2021)
36. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
37. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: ICML (2021)
38. Romero, J., Tzionas, D., Black, M.J.: Embodied hands: Modeling and capturing hands and bodies together. In: SIGGRAPH Asia (2017)
39. Rybkin, O., Daniilidis, K., Levine, S.: Simple and effective vae training with calibrated decoders. In: ICML (2021)
40. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: NTU RGB+D: A large scale dataset for 3D human activity analysis. In: CVPR (2016)
41. Streuber, S., Quiros-Ramirez, M.A., Hill, M.Q., Hahn, C.A., Zuffi, S., O'Toole, A., Black, M.J.: Body talk: Crowdshaping realistic 3D avatars with words. ACM TOG (2016)
42. Suveren-Erdogan, C., Suveren, S.: Teaching of basic posture skills in visually impaired individuals and its implementation under aggravated conditions. Journal of Education and Learning (2018)
43. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. CVPR (2015)
44. 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)

45. Yamada, T., Matsunaga, H., Ogata, T.: Paired recurrent autoencoders for bidirectional translation between robot actions and linguistic descriptions. IEEE RAL (2018)
46. Zhang, Y., Briq, R., Tanke, J., Gall, J.: Adversarial synthesis of human pose from text. In: GCPR (2020)
47. Zhou, Y., Barnes, C., Lu, J., Yang, J., Li, H.: On the continuity of rotation representations in neural networks. In: CVPR (2019)

Supplementary Material

We provide additional information about the data collection process in Section A. We give additional details on how to compute the different kinds of posecodes in Section B, and specify a list of those that are used in our work. In Section C, we elaborate on additional information about our automatic captioning pipeline and we specify the different versions of the captions we produced. Additional statistics about our PoseScript dataset are presented in Section D, and implementation details are given in Section E.

A Data collection process

Task instructions. A HIT (Human Intelligence Task) consists in writing the description of one given pose (in blue in the interface shown in Figure 3 of the main paper) precisely enough for the pose to be identified from its "discriminators" – the other similar poses, called discriminators (shown in grey in Figure 3). The instructions provided to the annotators are shown in Figure A.1.

Selection of pose discriminators. To select the pose discriminators for a given pose to be annotated, we compare it to the other poses of PoseScript. Similarity is measured using the distance between their pose embeddings, with an early version of our retrieval model. Discriminators are required to be the closest poses, while having at least 15 different posecode categorizations. This ensures that the selected poses share some semantic similarities with the pose to be annotated while having sufficient differences to be easily distinguished by the annotators. Discriminator examples are shown in Figure A.2.

Annotators qualifications. The HITs were initially made available for workers who:

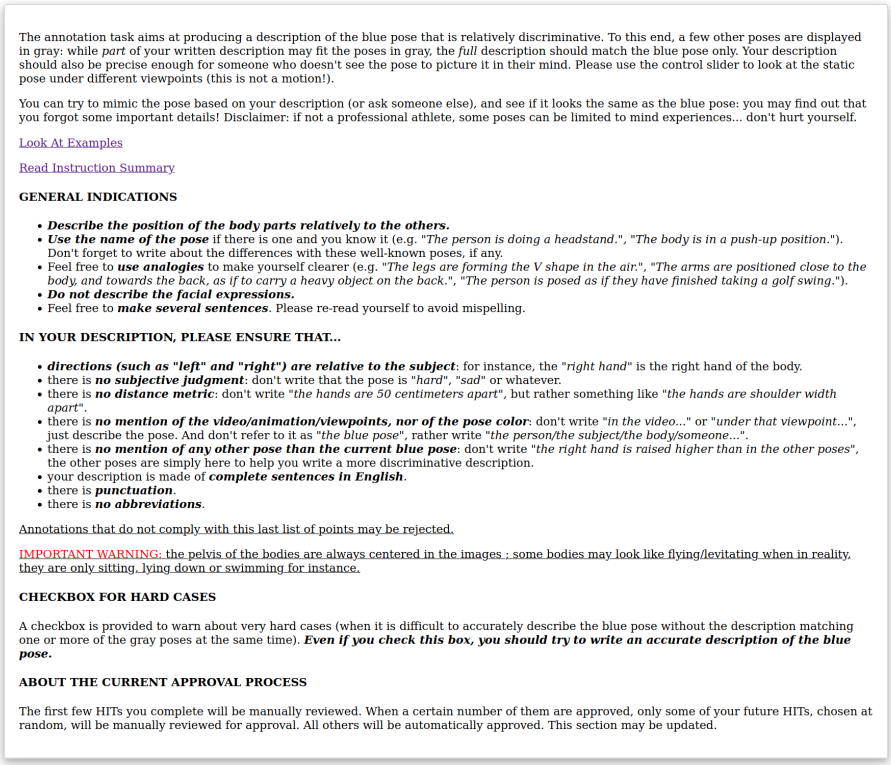
- live in English-speaking countries (USA, Canada, Australia, United-Kingdom, New Zealand),
- got at least 5000 of their HITs approved in the past,
- already have an approval rate larger or equal to 95%.

We manually read and evaluated close to 1000 HITs, based on the following main criteria:

- The description is 'complete', *i.e.*, nearly all the body parts are described.
- There is no left/right confusion (early mistakes were tolerated and manually curated, as writing while assuming the point of view of the body pose is not an easy task).
- The description refers to a static pose, and not to a motion, as some people mistook the rotation of the bodies for a motion.
- There is no distance metric.
- There is no subjective comment regarding the pose.

Based on these criteria, we qualified workers who produced excellent descriptions, and in a second step, HITs were only made available to them.

The time to complete a HIT was estimated to be 2-3 minutes. Each HIT was rewarded $0.50, based on the minimum wage in California for 2022. We additionally paid $2 bonus to qualified annotators for every 50 annotations.



The annotation task aims at producing a description of the blue pose that is relatively discriminative. To this end, a few other poses are displayed in gray: while *part* of your written description may fit the poses in gray, the *full* description should match the blue pose only. Your description should also be precise enough for someone who doesn't see the pose to picture it in their mind. Please use the control slider to look at the static pose under different viewpoints (this is not a motion!).

You can try to mimic the pose based on your description (or ask someone else), and see if it looks the same as the blue pose: you may find out that you forgot some important details! Disclaimer: if not a professional athlete, some poses can be limited to mind experiences... don't hurt yourself.

Look At Examples

Read Instruction Summary

GENERAL INDICATIONS

- ***Describe the position of the body parts relatively to the others.***
- ***Use the name of the pose*** if there is one and you know it (e.g. "*The person is doing a headstand.*", "*The body is in a push-up position.*"). Don't forget to write about the differences with these well-known poses, if any.
- Feel free to ***use analogies*** to make yourself clearer (e.g. "*The legs are forming the V shape in the air.*", "*The arms are positioned close to the body, and towards the back, as if to carry a heavy object on the back.*", "*The person is posed as if they have finished taking a golf swing.*").
- ***Do not describe the facial expressions.***
- Feel free to ***make several sentences***. Please re-read yourself to avoid mispelling.

IN YOUR DESCRIPTION, PLEASE ENSURE THAT...

- ***directions (such as "left" and "right") are relative to the subject***: for instance, the "*right hand*" is the right hand of the body.
- there is ***no subjective judgment***: don't write that the pose is "*hard*", "*sad*" or whatever.
- there is ***no distance metric***: don't write "*the hands are 50 centimeters apart*", but rather something like "*the hands are shoulder width apart*".
- there is ***no mention of the video/animation/viewpoints, nor of the pose color***: don't write "*in the video...*" or "*under that viewpoint...*", just describe the pose. And don't refer to it as "*the blue pose*", rather write "*the person/the subject/the body/someone...*".
- there is ***no mention of any other pose than the current blue pose***: don't write "*the right hand is raised higher than in the other poses*", the other poses are simply here to help you write a more discriminative description.
- your description is made of ***complete sentences in English***.
- there is ***punctuation***.
- there is ***no abbreviations***.

Annotations that do not comply with this last list of points may be rejected.

IMPORTANT WARNING: the pelvis of the bodies are always centered in the images : some bodies may look like flying/levitating when in reality, they are only sitting, lying down or swimming for instance.

CHECKBOX FOR HARD CASES

A checkbox is provided to warn about very hard cases (when it is difficult to accurately describe the blue pose without the description matching one or more of the gray poses at the same time). ***Even if you check this box, you should try to write an accurate description of the blue pose.***

ABOUT THE CURRENT APPROVAL PROCESS

The first few HITs you complete will be manually reviewed. When a certain number of them are approved, only some of your future HITs, chosen at random, will be manually reviewed for approval. All others will be automatically approved. This section may be updated.

Fig. A.1. Detailed task instructions provided to the annotators for the pose description task.

Fig. A.2. Example of discriminators. For the pose shown in blue (left column) to be annotated, we show the three discriminators that were selected in grey.

Statistics on the annotators. 159 different workers participated to the annotation process; 34 (21%) of them were qualified for the second annotation step. One annotator wrote 851 descriptions and five others close to or more than 300.

More human-written caption examples. To complement the human-written annotations shown in the main paper (left of Figure 2), we show in Figure A.3 additional examples of human-written captions.

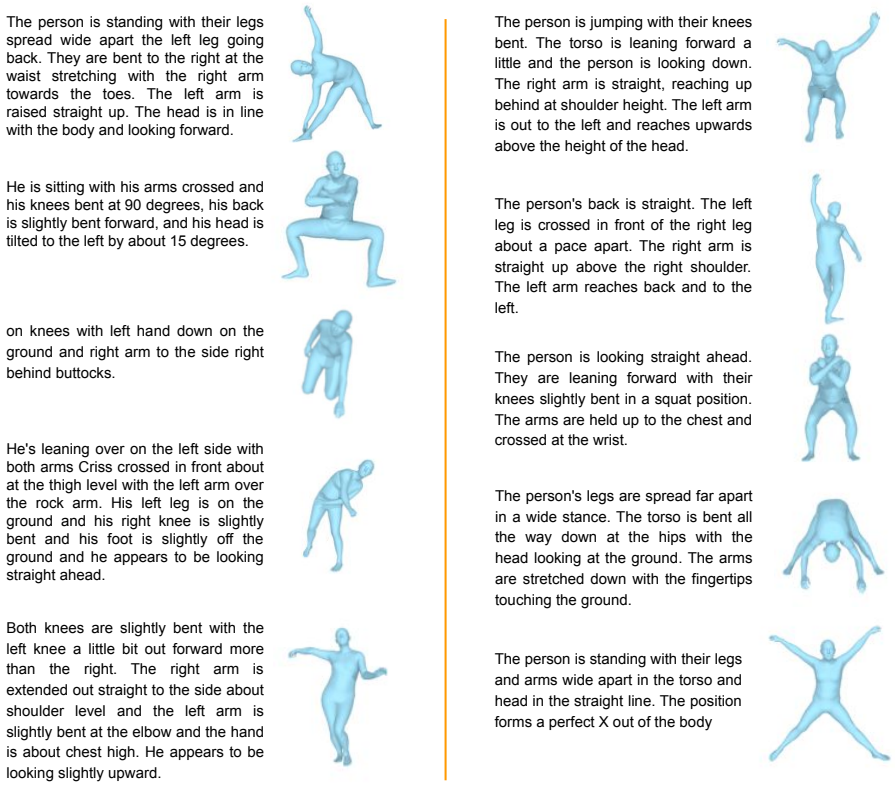


Fig. A.3. Additional examples of human-written captions from the PoseScript dataset.

B Posecodes

B.1 Computing posecodes

We detail here how the different kinds of posecodes are computed.

Elementary posecodes.

∘ *Angle posecodes* describe how a body part 'bends' around a joint j. Let a set of keypoints (i, j, k) where i and k are neighboring keypoints to j – for instance left shoulder, elbow and wrist respectively – and let p_l denote the position of keypoint l. The angle posecode is computed as the cosine similarity between vectors $v_{ji} = p_i - p_j$ and $v_{jk} = p_k - p_j$.

∘ *Distance posecodes* rate the $L2$-distance $\|v_{ij}\|$ between two keypoints i and j.

∘ *Posecodes on relative position* compute the difference between two sets of coordinates along a specific axis, to determine their relative positioning. A keypoint i is '`at the left of`' another keypoint j if $p_i^x > p_j^x$; it is '`above`' it if $p_i^y > p_j^y$; and '`in front of`' it if $p_i^z > p_j^z$.

∘ *Pitch & roll posecodes* assess the verticality or horizontality of a body part defined by two keypoints i and j. A body part is said to be '`vertical`' if the cosine similarity between $\frac{v_{ij}}{\|v_{ij}\|}$ and the unit vector along the y-axis is close to 0. A body part is said to be '`horizontal`' if it is close to 1.

∘ *Ground-contact posecodes* can be seen as specific cases of relative positioning posecodes along the y axis. They help determine whether a keypoint i is close to the ground by evaluating $p_i^y - \min_j p_j^y$. As not all poses are semantically in actual contact with the ground, we do not resort to these posecodes for systematic description, but solely for intermediate computations, to further infer super-posecodes for specific pose configurations.

Randomized binning step. As described above, each type of posecode is first associated to a value v (a cosine similarity angle or a distance), then binned into categories using predefined thresholds. In practice, hard deterministic thresholding is unrealistic as two different persons are unlikely to always have the same interpretation when the values are close to category thresholds, *e.g.* when making the distinction between '`spread`' and '`wide`'. Thus the categories are inherently ambiguous and to account for this human subjectivity, we randomize the binning step by defining a tolerable noise level η_τ on each threshold τ. We then categorize the posecode by comparing $v + \epsilon$ to τ, where ϵ is randomly sampled in the range $[-\eta_\tau, \eta_\tau]$. Hence, a given pose configuration does not always yield the exact same posecode categorization.

Super-posecodes are binary, and are not subject to the binning step. They only apply to a pose if all of the elementary posecodes they are based on possess the respective required posecode categorization.

B.2 List of posecodes

The list of the 77 elementary posecodes that are used in our work includes 4 angle posecodes, 22 distance posecodes, 34 posecodes describing relative positions (7 along the x-axis, 17 along the y-axis and 10 along the z-axis), 13 pitch & roll posecodes and 4 ground-contact posecodes. We specify the keypoints involved in the computation of each of these posecodes in Table B.1. Conditions for posecode categorizations (*i.e.*, thresholds applied to the measured angles and distances, with the corresponding random noise level) are indicated for each kind of posecode in Table B.2. Some of these elementary posecodes can be combined into super-posecodes. We list the 10 super-posecodes we currently consider in Table B.3, and indicate for each of them the different ways they can be produced from elementary posecodes.

Posecodes statistics. In Figures B.1, B.2, B.3, B.4, B.5, B.6 and B.7 we show posecode statistics obtained over the 20,000 poses of the PoseScript dataset. Specifically,

circle areas represent the proportion of poses satisfying the corresponding posecode categorization for the associated keypoints. We use the black and grey colors to denote categorizations that are ignored in the captioning process. A black circle area means that the corresponding pose configuration is too ambiguous (*e.g.* when the relative distance between two body parts is close to 0, making the detection of the body parts' relative position less obvious.). Grey circle areas denote trivial pose configurations (*e.g.* when a left body part is at the left of the associated right body part: this is the case most of the time). They correspond to posecode categorizations that apply to at least 60% of the poses. In contrast, posecode categorizations that describe less than 6% of the poses are defined as unskippable (*i.e.*, such pose information cannot be randomly discarded during the posecode selection process), and are colored in orange. All other available posecodes categorizations, in blue, are skippable (*i.e.*, such pose information can be randomly discarded during the posecode selection process). Equivalent information for super-posecodes is provided in Table B.3.

Most of the time, we follow statistics to consider posecode categorizations for pose description. In some specific cases, however, we are only interested in a subset of categorizations, and posecodes were only defined to retrieve such particular body pose information. This was done to infer super-posecodes later on (as for all ground-contact posecodes), or to bring in interesting semantics. For instance, distance posecodes involving one hand and another body part are only considered to inform about the position of the hand via the '`close`' category; indeed, while someone could describe the right hand as close to the left elbow, they are quite unlikely to point out that the right hand is wide apart from the left elbow. For the sake of completeness, we also present their statistics in the above-mentioned figures.

Angle posecodes
L-knee
R-knee
L-elbow
R-elbow

Ground-contact posecodes
L-knee
R-knee
L-foot
R-foot

Distance posecodes
L-elbow *vs.* R-elbow
L-hand *vs.* R-hand
L-knee *vs.* R-knee
L-foot *vs.* R-foot
L-hand *vs.* L-shoulder
L-hand *vs.* R-shoulder
R-hand *vs.* L-shoulder
R-hand *vs.* R-shoulder
L-hand *vs.* R-elbow
R-hand *vs.* L-elbow
L-hand *vs.* L-knee
L-hand *vs.* R-knee
R-hand *vs.* L-knee
R-hand *vs.* R-knee
L-hand *vs.* L-ankle
L-hand *vs.* R-ankle
R-hand *vs.* L-ankle
R-hand *vs.* R-ankle
L-hand *vs.* L-foot
L-hand *vs.* R-foot
R-hand *vs.* L-foot
R-hand *vs.* R-foot

Relative position posecodes
L-shoulder *vs.* R-shoulder (YZ)
L-elbow *vs.* R-elbow (YZ)
L-hand *vs.* R-hand (XYZ)
L-knee *vs.* R-knee (YZ)
R-foot *vs.* R-foot (XYZ)
neck *vs.* pelvis (XZ)
L-ankle *vs.* neck (Y)
R-ankle *vs.* neck (Y)
L-hip *vs.* L-knee (Y)
R-hip *vs.* R-knee (Y)
L-hand *vs.* L-shoulder (XY)
R-hand *vs.* R-shoulder (XY)
L-foot *vs.* L-hip (XY)
R-foot *vs.* R-hip (XY)
L-wrist *vs.* neck (Y)
R-wrist *vs.* neck (Y)
L-hand *vs.* L-hip (Y)
R-hand *vs.* R-hip (Y)
L-hand *vs.* torso (Z)
R-hand *vs.* torso (Z)
L-foot *vs.* torso (Z)
R-foot *vs.* torso (Z)

Pitch & roll posecodes
L-hip *vs.* L-knee
R-hip *vs.* R-knee
L-knee *vs.* L-ankle
R-knee *vs.* R-ankle
L-shoulder *vs.* L-elbow
R-shoulder *vs.* R-elbow
L-elbow *vs.* L-wrist
R-elbow *vs.* R-wrist
pelvis *vs.* L-shoulder
pelvis *vs.* R-shoulder
pelvis *vs.* neck
L-hand *vs.* R-hand
L-foot *vs.* R-foot

Table B.1. List of elementary posecodes. We provide the keypoints involved in each of the posecodes, for each type of elementary posecodes (angle, distance, relative position, pitch & roll or ground-contact). We grouped posecodes on relative positions for better readability, as some keypoints are studied along several axes (considered axes are indicated in parenthesis). Letters 'L' and 'R' stand for 'left' and 'right' respectively. Ignored, skippable and unskippable posecodes are shown in Figures B.1, B.2, B.3, B.4, B.5, B.6 and B.7.

Posecode type	Categorization	Condition
angle	`completely bent`	$v \pm 5 \leq 45$
	`almost completely bent`	$45 < v \pm 5 \leq 75$
	`bent at right angle`	$75 < v \pm 5 \leq 105$
	`partially bent`	$105 < v \pm 5 \leq 135$
	`slightly bent`	$135 < v \pm 5 \leq 160$
	`straight`	$v \pm 5 > 160$
distance	`close`	$v \pm 0.05 \leq 0.20$
	`shoulder width apart`	$0.20 < v \pm 0.05 \leq 0.40$
	`spread`	$0.40 < v \pm 0.05 \leq 0.80$
	`wide`	$v \pm 0.05 > 0.80$
relative position along the X axis	`at the right of`	$v \pm 0.05 \leq -0.15$
	`x-ignored`	$-0.15 < v \pm 0.05 \leq 0.15$
	`at the left of`	$v \pm 0.05 > -0.15$
relative position along the Y axis	`below`	$v \pm 0.05 \leq -0.15$
	`y-ignored`	$-0.15 < v \pm 0.05 \leq 0.15$
	`above`	$v \pm 0.05 > -0.15$
relative position along the Z axis	`behind`	$v \pm 0.05 \leq -0.15$
	`z-ignored`	$-0.15 < v \pm 0.05 \leq 0.15$
	`in front of`	$v \pm 0.05 > -0.15$
pitch & roll	`vertical`	$v \pm 5 \leq 10$
	`ignored`	$10 < v \pm 5 \leq 80$
	`horizontal`	$v \pm 5 > 80$
ground-contact	`on the ground`	$v \pm 0.05 \leq 0.10$
	`ground-ignored`	$v \pm 0.05 > 0.10$

Table B.2. Conditions for posecode categorizations. The right column provides the condition for a posecode to have the categorization indicated in the middle column. v represents the estimated value (an angle converted in degrees, or a distance in meters), while the number after the $\pm$ denotes the maximum noise value that can be added to v. Thresholds and noise levels depend only on the type of posecode.

Subject	Configuration	Eligibility	Production
torso	`horizontal`	🟠	*pitch & roll* (pelvis, L-shoulder) = `horizontal` *pitch & roll* (pelvis, R-shoulder) = `horizontal`
body	`bent left`	🔵	*relativePos Y* (L-ankle, neck) = `below` *relativePos X* (neck, pelvis) = `at left` *or* *relativePos Y* (R-ankle, neck) = `below` *relativePos X* (neck, pelvis) = `at left`
body	`bent right`	🔵	*relativePos Y* (L-ankle, neck) = `below` *relativePos X* (neck, pelvis) = `at right` *or* *relativePos Y* (R-ankle, neck) = `below` *relativePos X* (neck, pelvis) = `at right`
body	`bent backward`	🟠	*relativePos Y* (L-ankle, neck) = `below` *relativePos Z* (neck, pelvis) = `behind` *or* *relativePos Y* (R-ankle, neck) = `below` *relativePos Z* (neck, pelvis) = `behind`
body	`bent forward`	🔵	*relativePos Y* (L-ankle, neck) = `below` *relativePos Z* (neck, pelvis) = `front` *or* *relativePos Y* (R-ankle, neck) = `below` *relativePos Z* (neck, pelvis) = `front`
body	`kneel on left`	🟠	*relativePos Y* (L-knee, R-knee) = `below` *ground-contact* (L-knee) = `on the ground` *ground-contact* (R-foot) = `on the ground`
body	`kneel on right`	🟠	*relativePos Y* (L-knee, R-knee) = `above` *ground-contact* (R-knee) = `on the ground` *ground-contact* (L-foot) = `on the ground`
body	`kneeling`	🟠	*relativePos Y* (L-hip, L-knee) = `above` *relativePos Y* (R-hip, R-knee) = `above` *ground-contact* (L-knee) = `on the ground` *ground-contact* (R-knee) = `on the ground` *or* *angle* (L-knee) = `completely bent` *angle* (R-knee) = `completely bent` *ground-contact* (L-knee) = `on the ground` *ground-contact* (R-knee) = `on the ground`
hands	`shoulder width apart`	🟠	*distance* (L-hand, R-hand) = `shoulder width` *pitch & roll* (L-hand, R-hand) = `horizontal`
feet	`shoulder width apart`	🔵	*distance* (L-foot, R-foot) = `shoulder width` *pitch & roll* (L-foot, R-foot) = `horizontal`

Table B.3. List of super-posecodes. For each super-posecode, we indicate which body part(s) are subject to description (1st column) and their corresponding pose configuration (each super-posecode is given a unique category, indicated in the 2nd column). We additionally specify in the 3rd column whether the associated posecode is skippable for description, following the same color code as for elementary posecode statistics charts (🔵 : skippable; 🟠 : unskippable). Some super-posecodes can be produced by multiple sets of elementary posecodes: each set is separated by the word '*or*'. Letters 'L' and 'R' stand for 'left' and 'right' respectively.

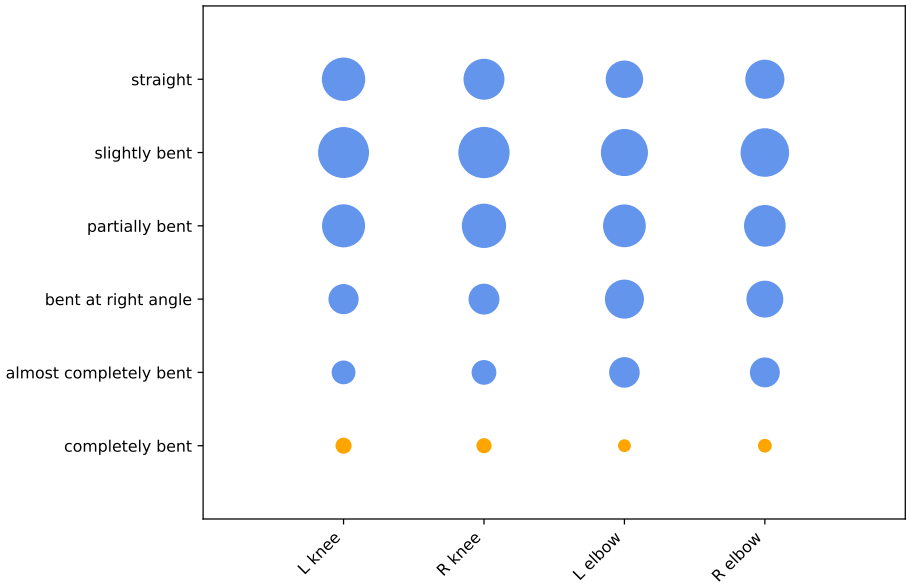


Fig. B.1. Statistics on categorizations of angle posecodes, obtained over all the poses of the PoseScript dataset. Letters 'L' and 'R' refer to left and right body parts respectively. The dot size varies with the proportion of poses that fit to the given categorization. Posecode categorizations used at captioning time are represented in orange (unskippable) and blue (skippable). For any keypoint, the posecode interpretation '`completely bent`' applies to less than 6% of the poses and is hence defined as unskippable.

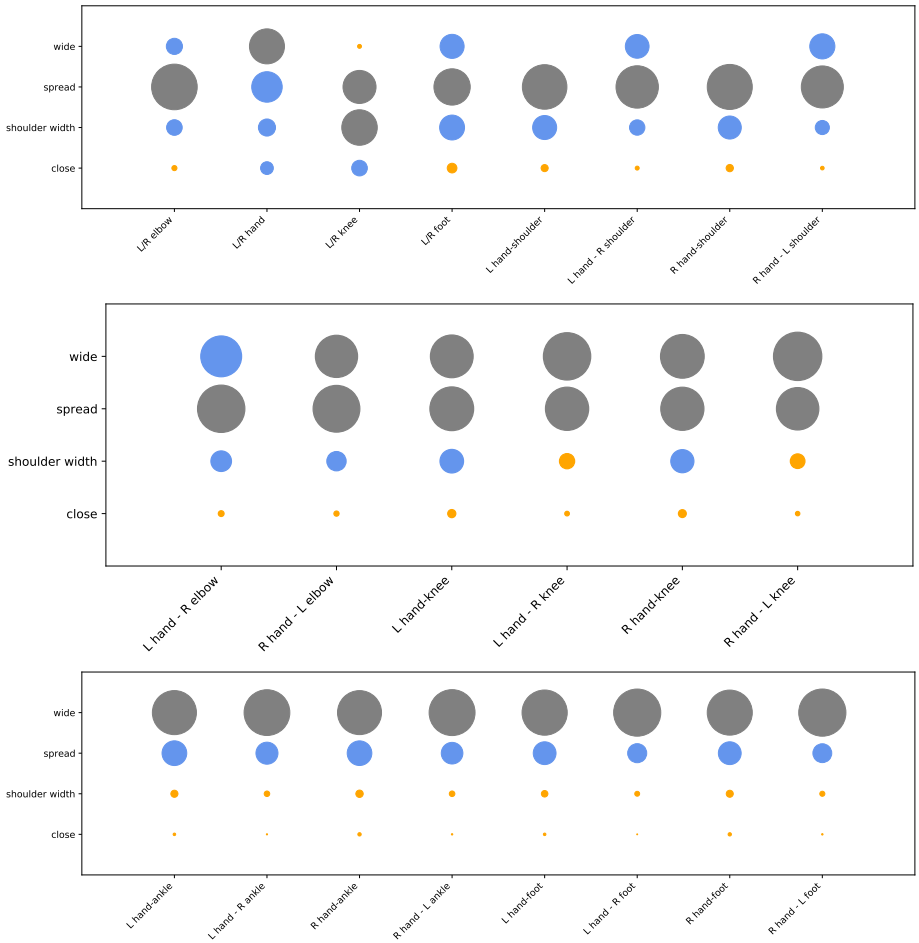


Fig. B.2. Statistics on categorizations of distance posecodes, obtained over all the poses of the PoseScript dataset. The first four columns of dots from the top block show distance posecodes between the left and right corresponding body parts; other columns of dots study the distance between a left or right body part and another left or right body part (when the side of the second body part is not specified, it is the same as for the first body part). Letters 'L' and 'R' refer to left and right body parts respectively. The dot size varies with the proportion of poses that fit to the given categorization. The dot color indicates unskippable (orange), skippable (blue), and ignored (grey) posecodes, based on their scarcity. In practice, when a distance posecode involves one of the hands only, we just consider the '`close`' categorization.

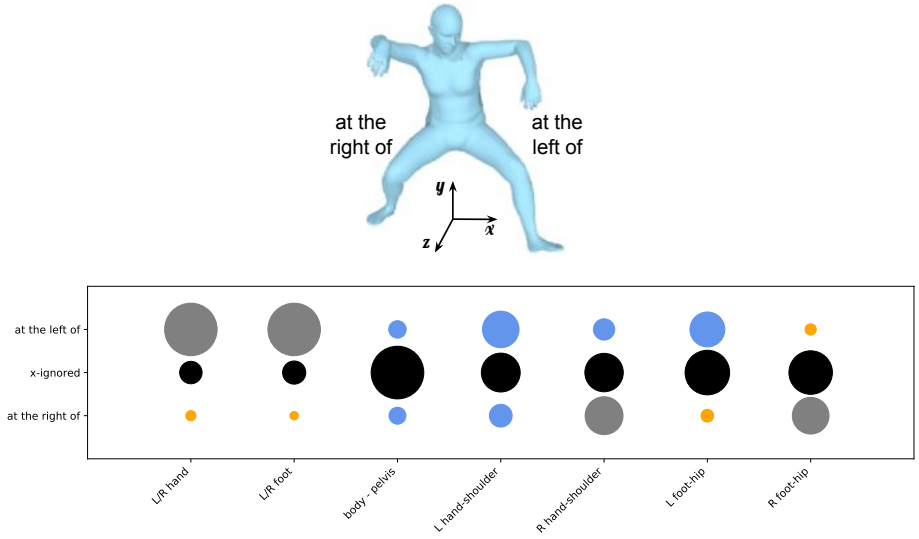


Fig. B.3. Statistics on categorizations of relative position posecodes along the X axis, obtained over all the poses of the PoseScript dataset. Letters 'L' and 'R' refer to left and right body parts respectively. When unspecified, pairs of body parts are from the same side of the body. The dot size varies with the proportion of poses that fit to the given categorization. The dot color indicates unskippable (orange), skippable (blue), and ignored (grey) posecodes, based on their scarcity. Black dots are ignored because of their inherent ambiguity. For instance, it appears that, for less than 6% of the poses (orange dots), body extremities (hand, foot) are crisscrossed. Such posecode categorizations are rare, and hence defined as unskippable. In some rare cases, dots representing similar relations between left-only body parts and right-only body parts are of different colors (note that dot sizes are still similar) because numbers fall close to the thresholds defining whether a relation should be unskippable/skippable/ignored. In such cases, the same rule is applied for right and left relations, *i.e.*, the left hand (resp. foot) being at the left of the left shoulder (resp. hip) is considered to be a gray dot.

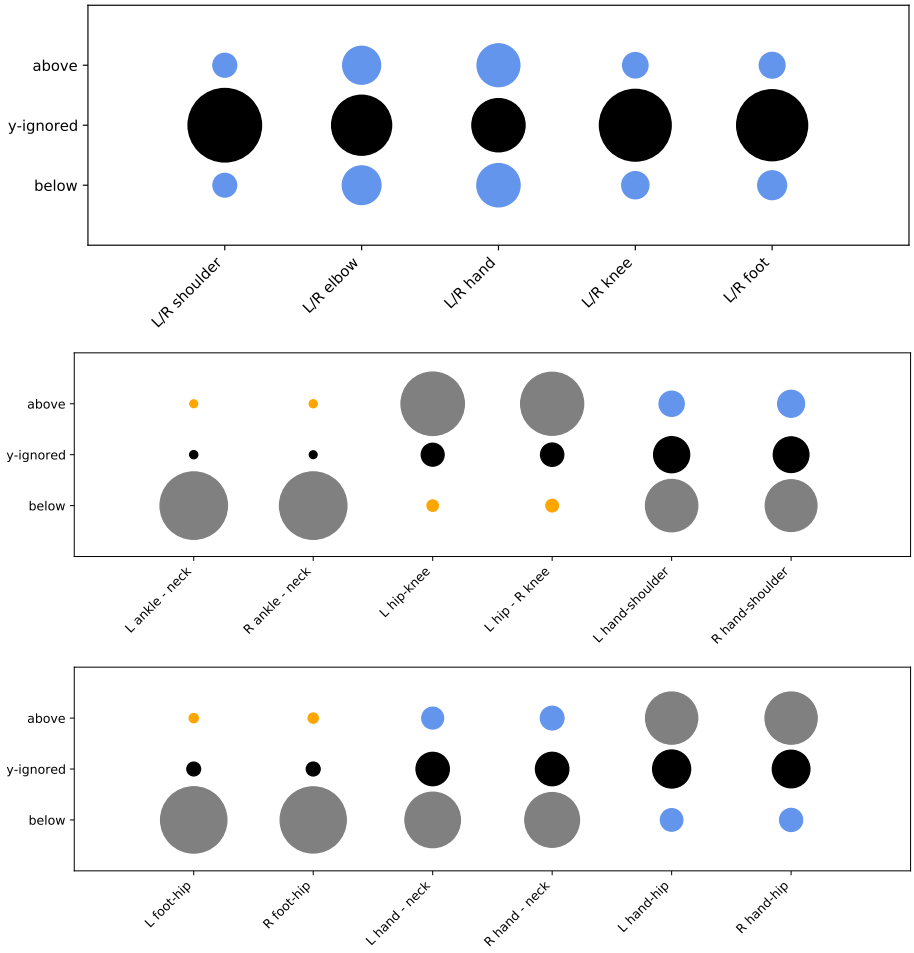


Fig. B.4. Statistics on categorizations of relative position posecodes along the Y axis, obtained over all the poses of the PoseScript dataset. The top block shows the relative position of the left body part with respect to the corresponding right body part. Following blocks study other relations; when unspecified, pairs of body parts are from the same side of the body. Letters 'L' and 'R' refer to left and right body parts respectively. The dot size varies with the proportion of poses that fit to the given categorization. The dot color indicates unskippable (orange), skippable (blue), and ignored (grey) posecodes, based on their scarcity. Black dots are ignored because of their inherent ambiguity. Note that the dataset is quite balanced regarding left-related and right-related relations (similar dot sizes). Some of these posecodes are considered only for super-posecode inference (*e.g.* L ankle - neck); in such cases the scarcity matters less than the provided information.

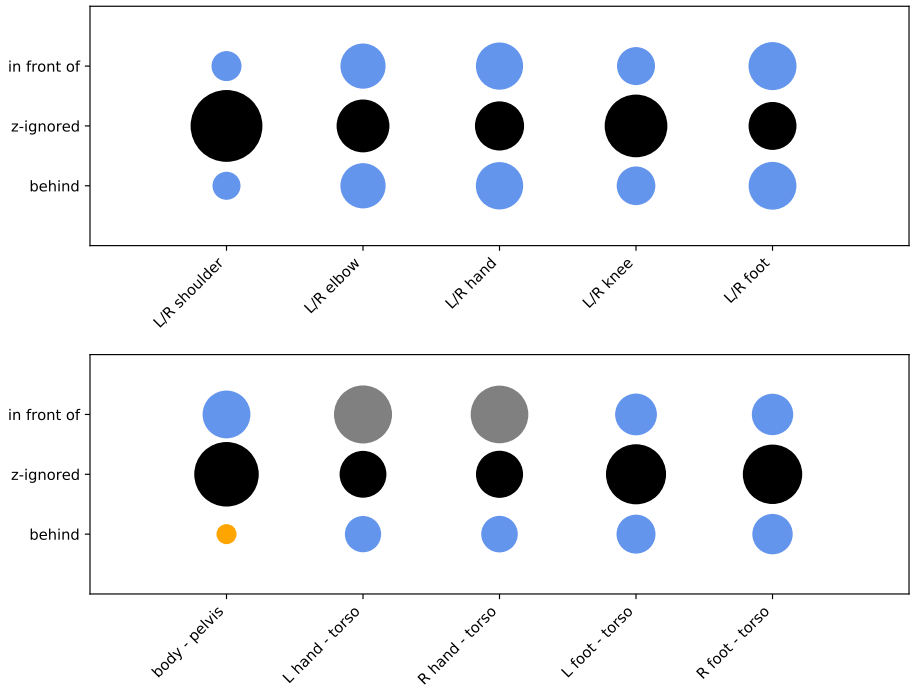


Fig. B.5. Statistics on categorizations of relative position posecodes along the Z axis, obtained over all the poses of the PoseScript dataset. The top block shows the relative position of the left body part with respect to the corresponding right body part; the lower block mainly presents the relative position of body extremities (hand/foot) with respect to the torso. The first column of the lower block actually studies the position of the neck with regard to the pelvis to further determine whether the body is bent (forward/backward). Letters 'L' and 'R' refer to left and right body parts respectively. The dot size varies with the proportion of poses that fit to the given categorization. The dot color indicates unskippable (orange), skippable (blue), and ignored (grey) posecodes, based on their scarcity. Black dots are ignored because of their inherent ambiguity.

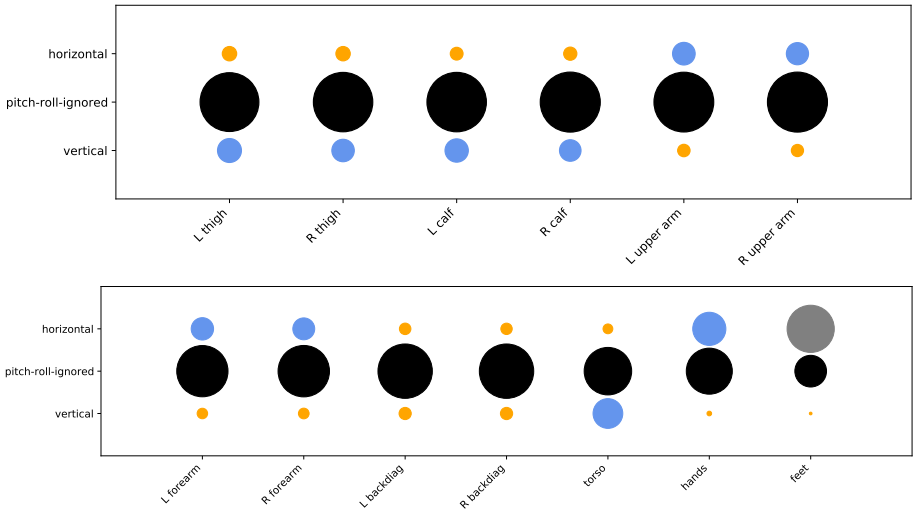


Fig. B.6. Statistics on categorizations of pitch & roll posecodes, obtained over all the poses of the PoseScript dataset. Letters 'L' and 'R' refer to left and right body parts respectively. The word 'backdiag' refers to the segment between the pelvis and the shoulder, 'hands' (resp. 'feet') to the segment between the two hands (resp. feet), and 'torso' to the segment between the neck and the pelvis. The dot size varies with the proportion of poses that fit to the given categorization. The dot color indicates unskippable (orange), skippable (blue), and ignored (grey) posecodes, based on their scarcity. Black dots are ignored because of their inherent ambiguity. Some of these posecodes are considered only for super-posecode inference (*e.g.* hands horizontality); in such cases the scarcity matters less than the information provided.

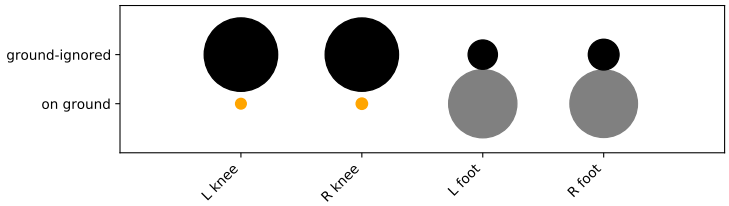


Fig. B.7. Statistics on categorizations of ground-contact posecodes, obtained over all the poses of the PoseScript dataset. Letters 'L' and 'R' refer to left and right body parts respectively. The dot size varies with the proportion of poses that fit to the given categorization. While the dot colors indicate different levels of scarcity, the '`on the ground`' categorization is used for all of these posecodes independently, for super-posecode inference only.

C More about the automatic captioning pipeline

In this section, we first detail the process used to generate the 6 automatic captions for each pose. Second, we present statistics about the captioning process. Third, we provide additional information about some steps of the captioning process.

C.1 Six versions of the automatically generated captions

All 6 captions, for each pose, were generated with the same pipeline. However, in order to propose captions with slightly different characteristics, we disabled some steps of the process when producing the different versions. Characteristics of the different caption versions are summarized in Table C.1. Specifically, steps that were deactivated include:

- Removing redundant posecodes based on ripple effect rules.
- Adding a sentence constructed from high-level pose annotations given by BABEL [35].
- Implicitness, *i.e.*, aggregating posecodes; omitting support keypoints (*e.g.* 'the right foot is behind the torso' does not turn into 'the right foot is in the back' when this step is deactivated) ; randomly referring to a body part by a substitute word (*e.g.* 'it'/'they', 'the other').
- Randomly skipping eligible posecodes for description.

Among all 20k poses of PoseScript, only 6,628 are annotated in BABEL and may benefit from an additional sentence in their automatic description. As 39% of PoseScript poses come from DanceDB, which was not annotated in BABEL, we additionally assign the 'dancing' label to those DanceDB-originated poses, for one variant of the automatic captions that already leverages BABEL auxiliary annotations (see Table C.1). This results in 14,435 poses benefiting from an auxiliary label. Figure C.1 shows an example of each caption version for a given pose in PoseScript.

Version	Random skip	Implicitness	Auxiliary labels	Ripple effect
A	✓	✓	✓(w/ dancing label)	✓
B	✓	✓	✓(w/ dancing label)	-
C	✓	✓	✓(w/o dancing label)	-
D	✓	✓	-	-
E	✓	-	-	-
F	-	-	-	-

Table C.1. Summary of the automatic caption versions. ✓ symbols indicate when characteristics apply to each caption version.

Human-written caption

The person is like doing a pose of hip-hop dance. The body is leaning slightly to the left with the thighs close to the floor and with supports on the right heel, the left foot and the left hand. The right leg is forward with the knee slightly bent. The left leg is almost complemtely bent. The left arm is stretched vertically, a bit backward. The right arm is forward and slightly up.

Automatic caption [version A]

The person is in a dancing pose. The right hand is wide apart from the left hand, towards the sky, the left elbow is in the back of the right. The left shoulder is lower than the right shoulder, the thighs and the right upper arm are horizontal. The right elbow is in l-shape while the left knee is bent sharply, the right knee is partially bent, the right foot is front.

Automatic caption [version B]

A person is making a dance pose. The left knee is right next to the right knee. It is bent sharply. The right knee is partly bent, the right foot is in the front and in front of the left foot, the right arm is raised above the left with the right elbow in l-shape, the body is bent on the left side while the right hand is reaching up and wide apart from the left hand and the left arm is in the back of the other and the left shoulder is further down than the other, the left hand is in their back and both thighs and the right upper arm are horizontal.

Automatic caption [version C]

The person is bent on the left side while their left foot is behind the other. Their right upper arm and their thighs are aligned horizontally with their right knee bent and near their left knee while their right foot is to the front with their left knee completely bent. Their left arm is further down than their right arm and their hands are wide apart while their left elbow is straight and their right shoulder is further up than the left. Their left hand is reaching backward. It is beneath their left hip. Their right elbow is forming a l shape, their right arm is in front of the other.

Automatic caption [version D]

Their right shoulder is raised above the left and their right elbow is bent at near a 90 degree angle with their right arm wide apart from the other, their right hand is towards the sky. Their left arm is further down than their right arm. It is located behind their right arm. Their thighs are aligned horizontally and their left knee is completely bent and their right knee is partially bent with their left foot behind the right. This person is angled towards the left while their left hand is back, lower than their left hip with their right foot in the front.

Automatic caption [version E]

Their left hand is in the back of their torso with their right elbow forming a l shape with their right knee approximately shoulder width apart from their left knee with their right thigh parallel to the ground and their left thigh horizontal with their right knee bent while their right hand is raised higher than their right shoulder. Their left foot is located behind their right foot with their right foot located in front of their torso and the figure bent over with their right shoulder raised over their left shoulder. Their left elbow is further down than their right elbow while their left knee is bent sharply while their left hand is in the back of their right hand. Their left elbow is straight with their body inclined to the left side while their right upper arm is parallel to the ground.

Automatic caption [version F]

His right hand is spread far apart from his left hand with his left elbow unbent. His left elbow is underneath his right elbow with his left hand lower than his left hip with his left elbow located behind his right elbow with his right upper arm horizontal while his left thigh is parallel to the floor and his right shoulder is lying over his left shoulder while his right thigh is flat, his body is leaning forwards. His right elbow is at right angle with his right hand raised over his neck. His left hand is in the back of his torso while his left foot is located behind his right foot while his right foot is in front of his torso. His right hand is over his right shoulder, his right knee is rather bent. His right hand is raised higher than his left hand with the figure leaning on his left side with his left knee bent sharply. His left knee is at the same level as his right knee with his right hand ahead of his left hand.

Fig. C.1. Captions from the different automatic versions for one pose in PoseScript.

C.2 Statistics about the captioning process

An average number of 303,495 'eligible' posecode categorizations were extracted from the 20,000 poses over the different caption versions (such 'eligible' posecodes are either represented by blue or orange dots in Figures B.1, B.2, B.3, B.4, B.5, B.6 and B.7 for elementary posecodes, and in Table B.3 for super-posecodes). During the posecode selection process, 42,981 of these were randomly skipped, and 6,286 were further removed to avoid redundancy. In practice, a bit less than 6% of the posecodes (17,570) are systematically kept for captioning due to being statistically discriminative (unskippable posecodes; orange dots). All caption versions were generated together in less than 6 minutes for the whole PoseScript dataset. Since the pose annotation task usually takes 2-3 minutes, it means we can generate 60k descriptions in the time it takes to manually write one.

Histograms about the number of posecodes used to generate the captions are presented in Figure C.2. Automatic captions are based on an average number of 13.4 posecodes. Besides, we observed that less than 0.1% of the poses had the exact same set of 87 posecode categorizations than another.

Histograms about about the number of words per automatic caption are additionally shown in Figure C.3.

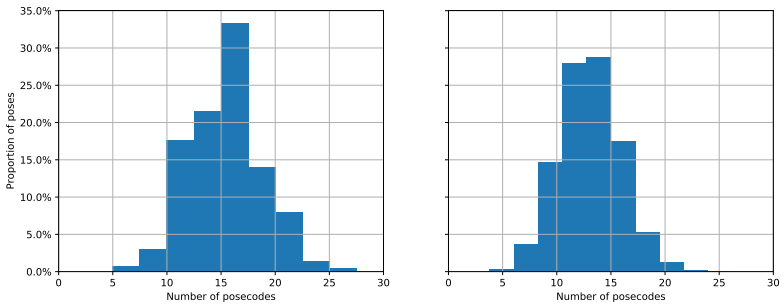


Fig. C.2. Histograms of the number of posecodes used per caption. The left histogram presents the number of posecodes for the the caption version `F`, which does not perform random skipping. The number of posecodes of each pose, in the right histogram, was averaged over the other 5 caption versions produced for each given pose (the ripple effect rules were not yet applied for version `A`). Random skip reduces the number of posecodes and thus impacts the length of the caption (see Figure C.3).

C.3 Miscellaneous details

Input to the pipeline. The process takes 3D joint coordinates of human-centric poses as input. These are inferred using the neutral body shape with default shape coefficients and a normalized global orientation along the y-axis. We use the resulting pose vector of dimension $N \times 3$ ($N = 52$ joints for the SMPL-H model [38]), augmented

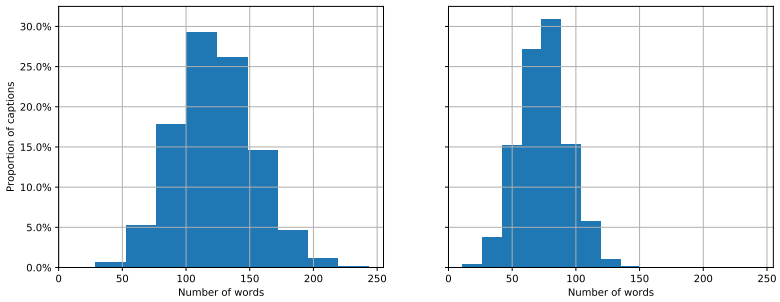


Fig. C.3. Histograms showing the number of words per automatic caption, for version `F` (left) and version `D` (right). An explanation of the length difference is that version `D` was obtained by randomly skipping some posecodes and generally aggregating them. Version `D` captions are assumed to be closer to what humans would write.

with a few additional keypoints, such as the left/right hands and the torso. They are deduced by simple linear combination of the positions of other joints, and are included to ease retrieval of pose semantics (*e.g.* a hand is in the back if it is behind the torso). Specifically:

- the hand keypoint is computed as the center between the wrist keypoint and the keypoint corresponding to the second phalanx of the hand's middle finger.
- the torso keypoint is computed as the average of the pelvis, the neck and the third spine keypoint.

What happens to posecodes contributing to super-posecodes?[5] There are three different outcomes for a posecode that contributes to a super-posecode:

- Some of the elementary posecodes are only *'support' posecodes*, and will never make it to the description alone: they only exist for computational purposes and need to be combined with other elementary posecodes to produce super-posecodes. For instance, to detect that the torso is parallel to the ground, we check that the two lines between the pelvis and each of the shoulders are horizontal. These two conditions are encoded via 'support' posecodes, which means that if the super-posecode is not produced because one of the two conditions is not satisfied, the second condition will not be transcribed in the caption: alone, it is meaningless.
- Some other posecodes can be considered as *'semi-support' posecodes*: they are discarded if the super-posecode they contribute to is successfully produced, but can make it to the description alone otherwise. For example, one way to detect that the body is kneeling is to check that both knees are completely bent, and in contact with the ground (otherwise the body could be in a squatting position). If all these conditions are met, the body is described as in a kneeling position and there is no need to further precise that the two knees are completely bent. If some of these conditions are not satisfied (*e.g.* the person is standing straight on their right foot),

[5] To reduce the verbosity of this paragraph, we refer to specific posecode categorizations as 'posecodes'.

the super-posecode is not produced, and conversely to a 'support' posecode, the 'semi-support' posecode 'the left knee is completely bent' is not discarded, as it carries important information.
- Remaining elementary posecodes, which contribute to super-posecodes but are neither 'support' nor 'semi-support' posecodes will make it to the description, no matter whether the super-posecodes they contribute to can be produced or not – unless they are skipped down the road, of course.

For more information about which posecodes are support and semi-support posecodes, please refer directly to the code.

How is the redundancy tackled in the captions?[6] Posecodes are numerous, and yet encode a single body pose. Between these constraints and those intrinsic to the human body (*e.g.* arms attached to the torso by the shoulders), information overlap arises quickly. In the automatic captions, redundancy is tackled in several ways: (1) posecodes summarized in aggregation rules are removed: information is passed on, not duplicated; (2) most of the posecodes contributing to super-posecodes are 'support' posecodes, that exist only for super-posecode inference and are removed afterwards; (3) redundant posecodes are further removed thanks to two kinds of ripple effect rules: (i) rules based on statistically frequent pairs and triplets of posecodes, and (ii) rules based on transitive relations between body parts. In details:

- **Relation-based rules** are mined automatically for each pose, and applied before any aggregation rule. For a given pose, if we have 3 posecodes telling that $a < b$, $b < c$ and $a < c$ (with a, b, and c being arbitrary body parts, and $<$ representing a relation of order such as *'below'*), then we keep only the posecodes telling that $a < b$ and $b < c$, as it is enough to infer the global relation $a < b < c$. For instance, with both *'L hand in front of torso'* and *'R hand behind torso'*, the posecode *'L hand in front of R hand'* is removed.
- **Statistics-based rules.** Let X and Y be two sets of posecodes. Let's write $p \sim Z$ a pose p that has all posecodes in a given set Z. We define a statistics-based rule $X \Rightarrow Y$ (X 'implies' Y) if

$$\frac{\sum_{p \in \text{PoseScript}} p \sim (X \cup Y)}{\sum_{p \in \text{PoseScript}} p \sim X} \geq \tau, \tag{3}$$

with $\tau = 1$ (ideally). In other words, if all the poses which have posecodes $X \cup Y$ can be summarized as having X only, then any pose that has X necessarily would have Y. This is a relatively safe assumption, as poses from PoseScript were selected to be as diverse as possible. We automatically mined statistics-based rules $X \Rightarrow Y$ such that $size(X) \leq 2$ and $size(Y) = 1$ with the following considerations:
  - the rule must involve eligible posecodes only, *i.e.*, posecodes that could make it to the description; trivial or ambiguous posecodes cannot be part of X or Y,
  - the rule must be symmetrically eligible for the left and right sides: the rule must work the same for the whole body,
  - the rule must affect at least 50 poses, *i.e.*, $\sum_{p \in \text{PoseScript}} p \sim X \geq 50$,
  - the rule must hold for at least 80% of the PoseScript poses when $size(X) = 2$ (*i.e.*, $\tau = 0.8$) and 70% when $size(X) = 1$ ($\tau = 0.7$).

[6] To reduce the verbosity of this paragraph, we refer to specific posecode categorizations as 'posecodes'.

We further reviewed all mined rules manually, to keep only the most meaningful and dispose of the following:

- rules where one of the posecodes in X could be considered an 'auxiliary' posecode, *i.e.*, a posecode used only to select a smaller set and make the denominator in equation (3) small enough to get past the selection threshold τ. This is particularly obvious when Y and one of the X posecodes are about the upper body while the other X posecode is about the lower body, for instance.
- rules with weak conditions, *e.g.* when X posecodes are providing conditions on left body parts relatively to right parts, to derive in Y a 'global' result on left body parts.

Statistics-based rules are computed before but applied after entity-based and symmetry-based aggregation rules; they consist in removing the Y posecodes if they still exist. For instance, with *'L hand above shoulder'*, *'R hand below hip'*, the posecode *'L hand above R hand'* is removed.

As a side note, annotators were found to repeat themselves in some captions.

Entity-based aggregation. We defined two very simple entities: the arm (formed by the elbow, and either the hand or the wrist; or by the upper-arm and the forearm) and the leg (formed by the knee, and either the foot or the ankle; or by the thigh and the calf).

Omitting support keypoints. We omit the second keypoint in the phrasing in those specific cases:

- a body part is compared to the torso,
- the hand is found 'above' the head,
- the hand (resp. foot) is compared to its associated shoulder (resp. hip), and is found either 'at the left of' or 'at the right of' of it. For instance, better than having 'the right hand is at the left of the left shoulder', which is quite tiresome, we would have *e.g.* 'the right hand is turned to the left'.

Use of negations in captions. We studied the use of negation in human-written captions: a bit less than 5% of them contain negations (*e.g.* '[close but] not touching' (20%), 'not quite/fully/completely/very' (15%), 'not bent' (10%)). Similar negations are easy to integrate in automatic caption templates. We did not include any as the proportion of negations in automatic captions would have been much greater than in human-written captions otherwise.

Context (environment/action) for pose generation. Context can be provided via another modality (*e.g.* an image) or freely expressed in natural language. We include BABEL [35] action labels in our automatic captions, and annotators were welcome to use analogies in their descriptions, *e.g. 'as if to climb a ladder'*. We primarily focus on learning explicit fine-grained relations between body parts (detailed & low-level pose indications). Physical environment constraints are beyond the scope of this work but make for an exciting future research direction.

Sensitivity to caption noise. We measure a variance of the mean recall below 0.5% when evaluating the retrieval model on 3 independent test sets obtained by generating different automatic captions per test pose, which shows robustness to changes in the query formulation. Some noise in human-written captions is inevitable but the generative model still produces reasonable results in practice.

D Dataset statistics

In this section, we provide some additional statistics about the PoseScript dataset.

Pose selection. Poses were sampled from 14,096 AMASS [26] sequences. Specifically, the first and last 25 frames of each sequence were skipped to avoid initialization poses (*e.g.* T-poses). Then we sampled one pose every 25 to avoid getting too similar poses (*i.e.*, consecutive poses). We used farther-sampling to further select 20,000 poses, which were found to come from 3,306 different sequences. Figure D.1 presents the AMASS sub-datasets from which come the poses selected for PoseScript. In particular, it appears that PoseScript poses come from almost all sequences of DanceDB and MPI_Limits that are available in AMASS; and that most of the poses in PoseScript actually come from DanceDB (39%), CMU (19%) and BioMotionLab (13%).

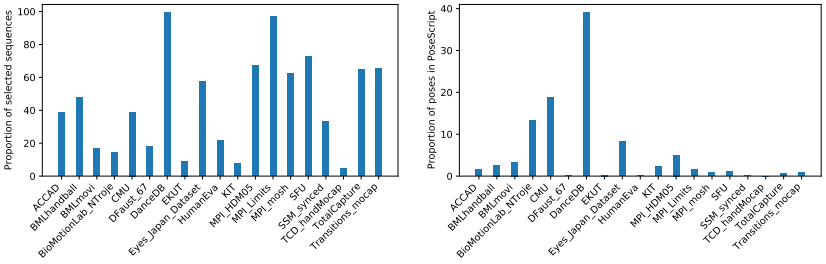


Fig. D.1. Origin of the selected poses. The left bar plot shows the proportion of sequences that are eventually used in PoseScript with respect to available sequences in AMASS. A sequence is 'used' if it provided at least one pose to PoseScript. The right bar plot shows the distribution of the PoseScript poses over the AMASS sub-datasets.

Sequence-based split. The selected poses were split into 3 subsets, such that poses from the same sequence are allocated to the same subset. As a result, the train set contains 14,004 poses from 2,183 different sequences, the validation set contains 2,025 poses from 369 other different sequences, and the test set contains 3,971 poses from 754 other different sequences.

Human-written captions. Histograms about the number of tokens and the number of words per human-written caption are presented in Figure D.2.

E Implementation details

Retrieval model. We use embeddings of size $d = 512$ and an initial loss temperature of $\gamma = 10$. GloVe word embeddings are 300-dimensional. The model is trained end to end for 500 epochs, using Adam [19], a batch size of 32 and an initial learning rate of 2.10^{-4} with a decay of 0.5 every 20 epochs.

Generative model. We follow exactly VPoser [28] for the pose encoder and decoder architectures (except that we use the 52 joints of SMPL-H [38]), and use the same text encoder as in the retrieval experiments. We train the models for 2000 epochs with

a batch size of 32, using the Adam optimizer, a learning rate of 10^{-4} (10^{-5} when finetuning) and a weight decay of 10^{-4}. The latent space has dimension 32.

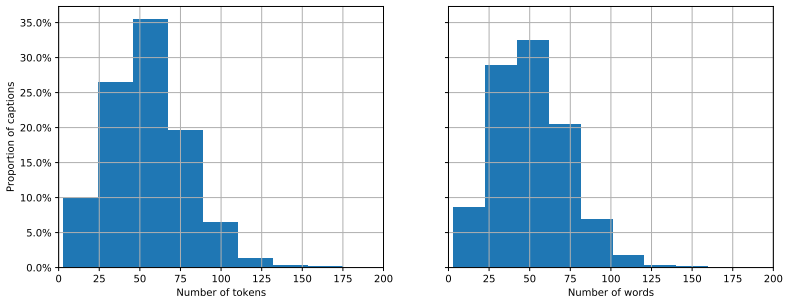


Fig. D.2. Histograms showing the number of tokens (left) **and the number of words** (right) **per human-written captions**. Tokens include words plus punctuation.