Programmatic Concept Learning for Human Motion Description and Synthesis Supplementary Materials

Sumith Kulal* Stanford University
Jiayuan Mao* MIT
Alex Aiken† Stanford University
Jiajun Wu† Stanford University

1. Details of Data Filtering and Model Fitting

As mentioned in Section 3.5 of the main paper, we use the alignments obtained from the description model to extract individual repetitions for all motion concepts from the training dataset. For each motion concept c, we obtain a dataset of occurrences $D_c = \{d_1, d_2, \cdots\}$ where each element d_i corresponds to a small segment composed of a sequence of spline curves $d_i = \{s_i^1, s_i^2, \cdots s_i^{l_i}\}$. Since this dataset D_c has been obtained from description model predictions, we perform two steps of filtering to generate cleaner data for learning our synthesis model.

Length Filtering. First, for each concept c, we compute the mode of the number of splines used in instances of this concept, denoted as $l_c^* = \textit{mode}\,(\{l_i\})$ and then filter out all d_i's whose number of splines is not l_c^*. It is possible to fit a synthesis model for each length and sample from each of these models but for simplicity we only consider models with a fixed number of primitives per class.

Similarity Filtering. We use the already annotated single repetition examples as the ground truth reference $\{g_1, g_2, \cdots\}$. We define the distance (*distance*) between d_i and g_j as the average L_2 distance between four points sampled at equal distance across all spline curves $\{s_i^1, s_i^2, \cdots s_i^{l_i}\}$ (note that the number of splines in d_i and g_j must match due to the first-step length filtering). Next, we filter out d_i if $\min_j \textit{distance}(d_i, g_j) > F$. Here, F is a hyperparameter which controls the error threshold. We choose $F = 8$ in our experiments.

Model Fitting. After these two steps of filtering, we now have dataset $D'_c = \{d'_1, d'_2, \cdots\}$ where each element d'_i is a sequence of spline curves $d'_i = \{t_i^1, t_i^2, \cdots t_i^{l_c^*}\}$. For each index ℓ with $1 \le \ell \le l_c^*$, we fit a simple Gaussian model over all occurrences

$$G_c^\ell \sim \mathcal{N}(\mu(t^l),\, \textit{cov\_f} \cdot \sigma(t^l)^2)$$

where *cov_f* is a hyperparameter that controls the variance of the generated motion. We set *cov_f* to 0.01 in our experiments and present ablation studies in the following section. To sample new motion instance $\bar{d}$, we sample each of G_c^l in sequence from l set to 1 to l_c^*.

2. Inference with Dynamic Programming

Given a primitive sequence $\bar{S}$, our goal in human motion description is to infer a label sequence $\bar{L}$. As described in Section 3.3 of the main paper, the inference of $\bar{L}$ is equivalent to finding the argmax of $p\left(\bar{L}|\bar{S}\right) = \sum_{\bar{C} \in \textit{uncompress}(\bar{L})} \prod_{t=1}^{2K-1} p\left(c_t|\bar{S}\right)$, where $p\left(c_t|\bar{S}\right)$ is the concept label sequence predicted by the neural network.

Since there are several possible alignments $\bar{C} \in \textit{uncompress}(\bar{L})$ for a given label sequence $\bar{L}$, the argmax could be quite expensive to compute in a brute-force manner. Hence, we use an efficient dynamic programming approach. The key idea is to compute prefix label sequences for prefix primitive sequences of $\bar{L}$ and merge different alignments that give the same output, which makes the inference and loss computation tractable. In practice, implementations of CTC manage this under-the-hood*.

3. Ablation of Covariance Factor

We study the effects of varying covariance factor (*cov_f*) on the generated motions. We present the results of our study in Figure 1. We observe that one can get motions with increased diversity and multimodality by increasing the *cov_f*. We observe that higher *cov_f* factor leads to less satisfactory visual quality, even if quantitatively there is only small drop of the recognition accuracy. Hence, we choose a lower default value of *cov_f* but this can be modified by the user as needed.

* and † indicate equal contribution. Project page: https://sumith1896.github.io/motion-concepts

*https://distill.pub/2017/ctc/

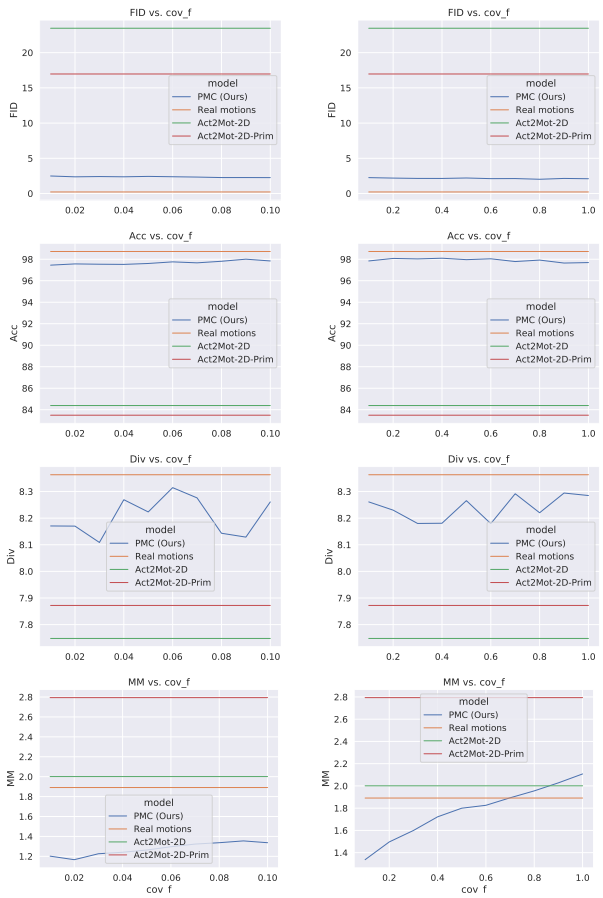

Figure 1. Ablating the covariance factor for evaluating action-conditioned motion synthesis. We compare the **FID** score, **Acc** (recognition accuracy), **Div** (variance across action classes) and **MM** (variance within action classes).

4. Ablation of Window Size

We study the effects of varying the window size (*WS*) on the motion description accuracy (SeqAcc). We present the results of our study in Figure 2. We observe that across all classes, the performance improves until it plateaus around *WS* equal to 9. We also present a breakdown on two specific classes: Jumping Jacks (JJ) and Torso Twists (TT). We observe that the easier JJ class achieves high accuracy for all values of *WS* while the harder TT class sees improvement with increasing *WS*. Intuitively, aggregating temporal context information helps up to certain lengths beyond which it provides no additional help.

5. Details of the Evaluation Metrics

We use standard metrics used in previous works to evaluate action-conditioned motion synthesis [3, 5] and controlled motion synthesis from descriptions [1, 2, 4]. In this section, we provide details of how these metrics were computed for our evaluations.

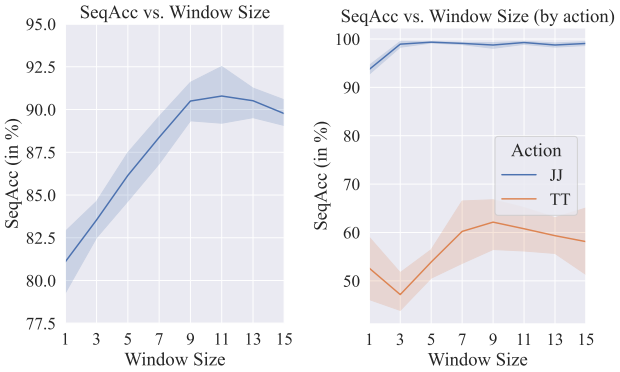

Figure 2. Ablating the window size for evaluating human motion description (SeqAcc). Left: ablation across all classes. Right: performance breakdown for two classes – Jumping Jacks (JJ, easy) and Torso Twists (TT, hard).

5.1. Action-Conditioned Motion Synthesis

We evaluate action-conditioned motion synthesis on four quantitative metrics that together try to capture the correctness and diversity of the synthesized motions. We compute the FID score, recognition accuracy (Acc), diversity (Div) and multimodality (MM). We train a simple RNN-based action classifier on the single repetition examples collected for each concept class. We use this for computing the recognition accuracy and also as a feature extractor for computing all other metrics. Closely following Guo et al. [3], we report the average of 20 independent runs for each metric.

FID score: We extract features of 1000 generated and real motions each (from test set with replacement). We then compute the FID between the generated distribution and real distribution. FID captures the overall quality of the generated motions.

Accuracy (Acc): We use the RNN-based action classifier to classify 1000 generated motions into classes. This indicates if the generated motions are recognized as the specified classes.

Diversity (Div): To capture the variance of generated motion across classes, we generate two subsets u and v of 200 motions each and extract features $\{u_1, u_2, \cdots, u_{200}\}$ and $\{v_1, v_2, \cdots, v_{200}\}$. We then compute the diversity metric defined as

$$\text{Div} = \frac{1}{200} \sum_{n=1}^{200} ||u_n - v_n||_2.$$

We use the RNN-based action classifier trained for the "Accuracy" measure (without the last linear layer for classification) as the feature extractor.

Multimodality (MM): To capture the variance of generated motion within classes, for each motion concept c, we generate two subsets u_c and v_c of 20 motions each and extract features $\{u_{c,1}, u_{c,2}, \cdots, u_{c,20}\}$ and $\{v_{c,1}, v_{c,1}, \cdots, v_{c,20}\}$. We then compute the multimodality metric defined as

$$\text{MM} = \frac{1}{C \times 20} \sum_{c=1}^{C} \sum_{n=1}^{20} ||u_{c,n} - v_{c,n}||_2$$

We use the RNN-based action classifier trained for the "Accuracy" measure (without the last linear layer for classification) as the feature extractor.

5.2. Controlled Motion Synthesis

We evaluate controlled motion synthesis on two metrics. For generated pose sequence $P = \{p_1, p_2, \cdots, p_T\}$ and ground truth pose sequence $\bar{P} = \{\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_T\}$, we compute the following two metrics.

Average Positional Error (APE): We define APE as the L_2 distance between the joint keypoints averaged across all joints over the time duration. Mathematically, it is defined as

$$\text{APE} = \frac{1}{J \times T} \sum_{t=1}^{T} \|p_t - \bar{p}_t\|_2.$$

Since the generated and ground-truth sequences could be of different lengths, we use Dynamic Time Warping (DTW) [6] to align these sequences.

Average Variance Error (AVE): We define AVE as the L_2 distance of variances of the generated motion with ground truth motion. We define variance of joint j as in a pose sequence $P = \{p_1, p_2, \cdots, p_T\}$ as:

$$\sigma(j) = \frac{1}{T-1} \sum_{t=1}^{T} \|p_t^j - \mu_p^j\|_2,$$

where μ_p^j is average location of joint j across the time duration, i.e., $\mu_p^j = \frac{1}{T} \sum_{t=1}^{T} p_t^j$. Similarly, we can define $\bar{\sigma}(j)$ for the groundtruth pose sequence $\bar{P}$. We then define AVE as:

$$\text{AVE} = \frac{1}{J} \sum_{j=1}^{J} \|\sigma(j) - \bar{\sigma}(j)\|_2.$$

6. Limitations and Societal Impact

Human motion has favorable structural properties such as constraints on the motion of keypoints and repetition that our methods exploit. Our method relies on primitive extraction which could be difficult in videos with occlusions or noisy pose detections. It is also not immediately clear how well these methods translate to motion of other entities.

Our research has potential positive societal impacts, with future applications in sports training and assistive technologies in the rehabilitation of disabled persons. On the other hand, like all other visual content generation methods, our method might be exploited by malicious users with potential negative impacts. In our code release, we will explicitly specify allowable uses of our system with appropriate licenses.

References

[1] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *2019 International Conference on 3D Vision (3DV)*, pages 719–728. IEEE, 2019. 2

[2] Anindita Ghosh, Noshaba Cheema, Cennet Oguz, Christian Theobalt, and Philipp Slusallek. Synthesis of compositional animations from textual descriptions. *arXiv preprint arXiv:2103.14675*, 2021. 2

[3] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 2

[4] Angela S Lin, Lemeng Wu, Rodolfo Corona, Kevin Tai, Qixing Huang, and Raymond J Mooney. Generating animated videos of human activities from natural language descriptions. *Learning*, 2018:1, 2018. 2

[5] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. *arXiv preprint arXiv:2104.05670*, 2021. 2

[6] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007. 3