A Stochastic Conditioning Scheme for Diverse Human Motion Prediction*

Sadegh Aliakbarian[†1,2,4], Fatemeh Sadat Saleh[†1,2], Mathieu Salzmann[3], Lars Petersson[1,4], Stephen Gould[1,2]
[1]Australian National University, [2]ACRV, [3]CVLab, EPFL, [4]Data61, CSIRO
{fname.lname}@anu.edu.au, methieu.salzmann@epfl.ch, lars.petersson@data61.csiro.au
https://mix-and-match.github.io/

Abstract

Human motion prediction, the task of predicting future 3D human poses given a sequence of observed ones, has been mostly treated as a deterministic problem. However, human motion is a stochastic process: Given an observed sequence of poses, multiple future motions are plausible. Existing approaches to modeling this stochasticity typically combine a random noise vector with information about the previous poses. This combination, however, is done in a deterministic manner, which gives the network the flexibility to learn to ignore the random noise. Alternatively, in this paper, we propose to stochastically combine the root of variations with previous pose information, so as to force the model to take the noise into account. We exploit this idea for motion prediction by incorporating it into a recurrent encoder-decoder network with a conditional variational autoencoder block that learns to exploit the perturbations. Our experiments on two large-scale motion prediction datasets demonstrate that our model yields high-quality pose sequences that are much more diverse than those from state-of-the-art stochastic motion prediction techniques.

1. Introduction

Human motion prediction aims to forecast the sequence of future poses of a person given past observations of such poses. To achieve this, existing methods typically rely on recurrent neural networks (RNNs) that encode the person's motion [28, 14, 37, 22, 5, 31, 32]. While they predict reasonable motions, RNNs are deterministic models and thus cannot account for the highly stochastic nature of human motion; given the beginning of a sequence, multiple, diverse futures are plausible. To correctly model this, it is therefore critical to develop algorithms that can learn the *multiple modes* of human motion, even when presented with only deterministic training samples.

*This research was supported by the Australian Government through the Australian Research Council (ARC).

†Equal contribution.

Recently, several attempts have been made at modeling the stochastic nature of human motion [40, 5, 37, 22, 26]. These methods rely on sampling a random vector that is then combined with an encoding of the observed pose sequence. In essence, this combination is similar to the conditioning of generative networks; the resulting models aim to generate an output from a random vector while taking into account additional information about the content.

While standard conditioning strategies, i.e., concatenating the condition to the latent variable, may be effective for many tasks, as in [41, 21, 11, 10, 4, 23], they are ill-suited for motion prediction. The reason is the following: In other tasks, the conditioning variable only provides auxiliary information about the output to produce, such as the fact that a generated face should be smiling. By contrast, in motion prediction, it typically contains the core signal to produce the output, i.e., the information about the previous poses. We empirically observed that, since the prediction model is trained using deterministic samples (i.e., one condition per sample), it can then simply learn to ignore the random vector and still produce a meaningful output based on the conditioning variable only. In other words, the model can ignore the root of variations, and thus essentially become deterministic. This problem was discussed in [6] in the context of unconditional text generation, and we identified it in our own motion prediction experiments.

We introduce a simple yet effective approach to counteracting this loss of diversity and thus to generating truly diverse future pose sequences. At the heart of our approach lies the idea of *Mix-and-Match* perturbations: Instead of combining a noise vector with the conditioning variables in a deterministic manner, we randomly select and perturb a subset of these variables. By randomly changing this subset at every iteration, our strategy prevents training from identifying the root of variations and forces the model to take it into account in the generation process. Consequently, as supported by our experiments, our approach produces not only high-quality predictions but also truly diverse ones.

In short, our contributions are *(i)* a novel way of imposing diversity into conditional VAEs, called *Mix-and-Match*

perturbations; *(ii)* a new motion prediction model capable of generating multiple likely future pose sequences from an observed motion; *(iii)* a new set of evaluation metrics for quantitatively measuring the quality and the diversity of generated motions, thus facilitating the comparison of different stochastic approaches; and *(iv)* a curriculum learning paradigm for training generative models that use Mix-and-Match perturbation as the stochastic conditioning scheme. Despite its simplicity, curriculum learning of variation is essential to achieve optimal performance in case of imposing large variations.

2. Related Work

Deterministic Motion Prediction. Most motion prediction approaches are based on *deterministic* models [32, 31, 14, 18, 28, 15, 12, 13, 27], casting motion prediction as a regression task where only one outcome is possible given the observations. Due to the success of RNN-based methods at modeling sequence-to-sequence learning problems, many attempts have been made to address motion prediction within a recurrent framework [28, 14, 37, 22, 5, 31, 32]. Typically, these approaches try to learn a mapping from the observed sequence of poses to the future sequence. Another group of study addresses this problem within feed-forward models [27, 24, 7], either with fully-connected [7], convolutional [24], or more recently, graph neural networks [27]. While a deterministic approach may produce accurate predictions, it fails to reflect the stochastic nature of human motion, where multiple plausible outcomes can be highly likely for a single given series of observations. Modeling this diversity is the topic of this paper, and we therefore focus the discussion below on the other methods that have attempted to do so.

Stochastic Motion Prediction. The general trend to incorporate variations in the predicted motions consists of combining information about the observed pose sequence with a random vector. In this context, two types of approaches have been studied: The techniques that directly incorporate the random vector into the RNN decoder, e.g., as in GANs, and those that make use of an additional Conditional Variational Autoencoder (CVAE) [36] to learn a latent variable that acts as the root of variation.

In the first class of methods, [26] sample a random vector $z_t \sim \mathcal{N}(0, I)$ at each time step and add it to the pose input to the RNN decoder. By relying on different random vectors at each time step, however, this strategy is prone to generating discontinuous motions. To overcome this, [22] make use of a single random vector to generate the entire sequence. This vector is both employed to alter the initialization of the decoder and concatenated with a pose embedding at each iteration of the RNN. By relying on concatenation as a mean to fuse the condition and the random vector, these two methods contain parameters that are specific to the random vector, and thus give the model the flexibility to ignore this information. In [5], instead of using concatenation, the random vector is added to the hidden state produced by the RNN encoder. While addition prevents having parameters that are specific to the random vector, this vector is first transformed by multiplication with a parameter matrix, and thus can again be zeroed out so as to remove the source of diversity, as we observe empirically in Section 4.2.

The second category of stochastic methods introduce an additional CVAE between the RNN encoder and decoder. This allows them to learn a more meaningful transformation of the noise, combined with the conditioning variables, before passing the resulting information to the RNN decoder. In this context, [37] propose to directly use the pose as conditioning variable. As will be shown in our experiments, while this approach is able to maintain some degree of diversity, albeit less than ours, it yields motions of lower quality because of its use of independent random vectors at each time step. In [8], an approach similar to that of [37] is proposed, but with one CVAE per limb. As such, this method suffers from the same discontinuity problem as [37, 26]. Finally, instead of perturbing the pose, the recent work of [40] uses the RNN decoder hidden state as conditioning variable in the CVAE, concatenating it with the random vector. While this approach generates high-quality motions, it suffers from the fact that the CVAE decoder gives the model the flexibility to ignore the random vector.

Ultimately, both classes of methods suffer from the fact that they allow the model to ignore the random vector, thus relying entirely on the conditioning information to generate future poses. Here, we introduce an effective way to maintain the root of diversity by randomizing the combination of the random vector with the conditioning variable.

3. Proposed Method

In this section, we first introduce our *Mix-and-Match* approach to introducing diversity in CVAE-based motion prediction. We then describe the motion prediction architecture we used in our experiments and propose a novel evaluation metric to quantitatively measure the diversity and quality of generated motions.

3.1. Mix-and-Match Perturbation

The main limitation of prior work in the area of stochastic motion modeling, such as [37, 5, 40], lies in the way they fuse the random vector with the conditioning variable, i.e., RNN hidden state or pose, which causes the model to learn to ignore the randomness and solely exploit the deterministic conditioning information to generate motion. To overcome this, we propose to make it harder for the model to decouple the random variable from the deterministic information. Specifically, we observe that the way the random variable and the conditioning one are combined in existing

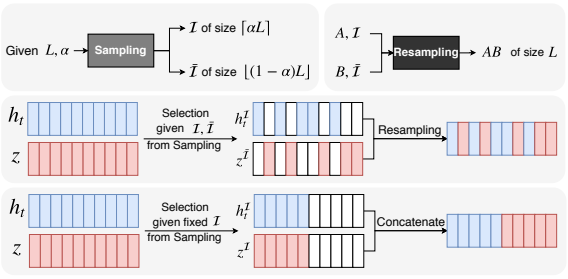

Figure 1. Mix-and-Match perturbation. **(Top)** Illustration of the *Sampling* operation (left) and of the *Resampling* one (right). Given a sampling rate α and a vector length L, the Sampling operation samples $\lceil \alpha L \rceil$ indices, say $\mathcal{I}$. The complementary, unsampled indices are denoted by $\bar{\mathcal{I}}$. Then, given two L-dimensional vectors and the corresponding $\lceil \alpha L \rceil$ and $\lfloor (1-\alpha)L \rfloor$ indices, the Resampling operation mixes the two vectors to form a new L-dimensional one. **(Middle)** Example of Mix-and-Match perturbation. **(Bottom)** Example of perturbation by concatenation, as in [40]. Note that, in Mix-and-Match perturbations, sampling is stochastic; the indices are sampled uniformly randomly for each mini-batch. By contrast, in [40], sampling is deterministic, and the indices in $\mathcal{I}$ are fixed and correspond to $\mathcal{I} = \{1, \ldots, \frac{L}{2}\}$.

methods is deterministic. We therefore propose to make this process stochastic.

Similarly to [40], we propose to make use of the hidden state as the conditioning variable and generate a perturbed hidden state by combining a part of the original hidden state with the random vector. However, as illustrated in Fig. 1, instead of assigning predefined, deterministic indices to each piece of information, such as the first half for the hidden state and the second one for the random vector, we assign the values of the hidden state to *random* indices and the random vector to the complementary ones.

More specifically, as depicted in Fig. 1, a mix-and-match perturbation takes two vectors of size L as input, say h_t and z, and combines them in a stochastic manner. To this end, it relies on two operations. The first one, called *Sampling*, chooses $\lceil \alpha L \rceil$ indices uniformly at random among the L possible values, given a sampling rate $0 \leq \alpha \leq 1$. Let us denote by $\mathcal{I} \subseteq \{1, \ldots, L\}$, the resulting set of indices and by $\bar{\mathcal{I}}$ the complementary set. The second operation, called *Resampling*, then creates a new L-dimensional vector whose values at indices in $\mathcal{I}$ are taken as those at corresponding indices in the first input vector and the others at the complementary indices, of dimension $\lfloor (1-\alpha)L \rfloor$, in the second input vector.

3.2. M&M Perturbation for Motion Prediction

Let us now describe the way we use our mix-and-match perturbation strategy for motion prediction. To this end, we first discuss the network we rely on during inference, and then explain our training strategy.

Inference. The high-level architecture we use at inference time is depicted by Fig. 2 (Top). It consists of an RNN encoder that takes t poses $x_{1:t}$ as input and outputs an L-dimensional hidden vector h_t. A random $\lceil \alpha L \rceil$-dimensional portion of this hidden vector, $h_t^{\mathcal{I}}$, is then combined with an $\lfloor (1-\alpha)L \rfloor$-dimensional random vector $z \sim \mathcal{N}(0, I)$ via our mix-and-match perturbation strategy. The resulting L-dimensional output is passed through a small neural network (i.e., *ResBlock2* in Fig. 2) that reduces its size to $\lceil \alpha L \rceil$, and then fused with the remaining $\lfloor (1-\alpha)L \rfloor$-dimensional portion of the hidden state, $h_t^{\bar{\mathcal{I}}}$. This, in turn, is passed through the VAE decoder to produce the final hidden state h_z, from which the future poses $x_{t+1:T}$ are obtained via the RNN decoder.

Training. During training, we aim to learn both the RNN parameters and the CVAE ones. Because the CVAE is an *auto*encoder, it needs to take as input information about future poses. To this end, we complement our inference architecture with an additional RNN future encoder, yielding the training architecture depicted in Fig. 2 (Bottom). Note that, in this architecture, we incorporate an additional mix-and-match perturbation that fuses the hidden state of the RNN past encoder h_t with that of the RNN future encoder h_T and forms h_{tT}^p. This allows us to condition the VAE encoder in a manner similar to the decoder. Note that, for each mini batch, we use the same set of sampled indices for all mix-and-match perturbation steps throughout the network. Furthermore, following the standard CVAE strategy, during training, the random vector z_p is sampled from the approximate posterior distribution $\mathcal{N}(\mu_\theta(x), \Sigma_\theta(x))$, whose mean $\mu_\theta(x)$ and covariance matrix $\Sigma_\theta(x)$ are produced by the CVAE encoder with parameters θ. This, in practice, is done by the reparameterization technique [20]. Note that, during inference, $z_p = \epsilon \sim \mathcal{N}(0, I)$ since we do not have access to x, hence to $\mu_\theta(x)$ and $\Sigma_\theta(x)$.

To learn the parameters of our model, we rely on the availability of a dataset $D = \{X_1, X_2, ..., X_N\}$ containing N videos X_i depicting a human performing an action. Each video consists of a sequence of T poses, $X_i = \{x_i^1, x_i^2, ..., x_i^T\}$, and each pose comprises J joints forming a skeleton, $x_i^t = \{x_{i,1}^t, x_{i,2}^t, ..., x_{i,J}^t\}$. The pose of each joint is represented as a 4D quaternion. Given this data, we train our model by minimizing a loss function of the form

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \Big(\mathcal{L}_{rot}(X_i) + \mathcal{L}_{skl}(X_i) \Big) + \lambda \mathcal{L}_{prior} \ . \quad (1)$$

The first term in this loss compares the output of the network with the ground-truth motion using the squared loss. That is,

$$\mathcal{L}_{rot}(X_i) = - \sum_{k=t+1}^{T} \sum_{j=1}^{J} \| \hat{x}_{i,j}^k - x_{i,j}^k \|^2 \ , \quad (2)$$

where $\hat{x}_{i,j}^k$ is the predicted 4D quaternion for the j^{th} joint at time k in sample i, and $x_{i,j}^k$ the corresponding ground-

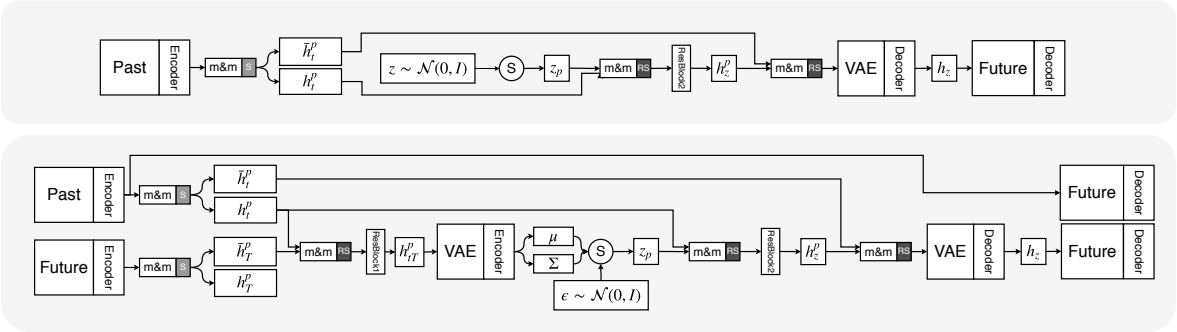

Figure 2. Overview of our approach. **(Top): Overview of the model during inference**. During inference, given past information and a random vector sampled from a Normal distribution, the model generates new motions. **(Bottom): Overview of the model during training.** During training, we use a future pose autoencoder with a CVAE between the encoder and the decoder. The RNN encoder-decoder network mapping the past to the future then aims to generate good conditioning variables for the CVAE.

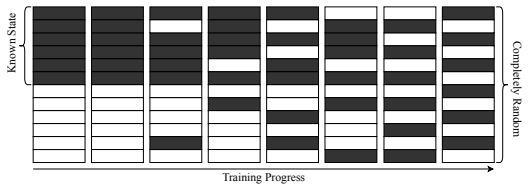

Figure 3. Example of curriculum perturbation of the hidden state.

truth one. The main weakness of this loss is that it treats all joints equally. However, when working with angles, some joints have a much larger influence on the pose than others. For example, because of the kinematic chain, the pose of the shoulder affects that of the rest of the arm, whereas the pose of the wrists has only a minor effect. To take this into account, we define our second loss term as the error in 3D space. That is,

$$\mathcal{L}_{skl}(X_i) = -\sum_{k=t+1}^{T}\sum_{j=1}^{J} \|\hat{p}_{i,j}^k - p_{i,j}^k\|^2 \,, \qquad (3)$$

where $\hat{p}_{i,j}^k$ is the predicted 3D position of joint j at time k in sample i and $p_{i,j}^k$ the corresponding ground-truth one. These 3D positions can be computed using forward kinematics, as in [32, 31]. Note that, to compute this loss, we first perform a global alignment of the predicted pose and the ground-truth one by rotating the root joint to face [0, 0, 0]. Finally, following standard practice in training VAEs, we define our third loss term as the KL divergence

$$\mathcal{L}_{prior} = -KL\Big(\mathcal{N}(\mu_\theta(x), \Sigma_\theta(x)))\|\mathcal{N}(0, I)\Big)$$

$$= -\frac{1}{2}\sum_{j=1}^{d}\Big(1 + \log(\sigma_\theta(x)_j^2) - \mu_\theta(x)_j^2 - \sigma_\theta(x)_{c_j}^2\Big) \,. \quad (4)$$

where $\Sigma_\theta(x) = diag(\sigma_\theta(x)^2)I$ and d is the length of the diagonal of the covariance matrix. In practice, since our VAE appears within a recurrent model, we weigh $\mathcal{L}_{prior}$ by a function λ corresponding to the KL annealing weight of [6]. We start from $\lambda = 0$, forcing the model to encode as much information in z as possible, and gradually increase it to $\lambda = 1$, following a logistic curve.

3.3. Curriculum Learning of Variation

The parameter α in our mix-and-match perturbation scheme determines a trade-off between stochasticity and motion quality. The larger α, the larger the portion of the original hidden state that will be perturbed. Thus, the model incorporates more randomness and less information from the original hidden state. As such, given a large α, it becomes harder for the model to deliver motion information from the observation to the future representation since a large portion of the hidden state is changing randomly. In particular, we observed that training becomes unstable if we use a large α from the beginning, with the motion-related loss terms fluctuating while the prior loss $\mathcal{L}_{prior}$ quickly converges to zero. To overcome this while still enabling the use of sufficiently large values of α to achieve high diversity, we introduce the curriculum learning strategy depicted by Fig. 3. In essence, we initially select $\lceil \alpha L \rceil$ indices in a deterministic manner and gradually increase the randomness of these indices as training progresses. More specifically, given a set of $\lceil \alpha L \rceil$ indices, we replace c indices from the sampled ones with the corresponding ones from the remaining $\lfloor (1-\alpha)L \rfloor$ indices. Starting from $c = 0$, we gradually increase c to the point where all $\lceil \alpha L \rceil$ indices are sampled uniformly randomly. More details, including the pseudo-code of this approach, are provided in the supplementary material. This strategy helps the motion decoder to initially learn and incorporate information about the observations (as in [40]), yet, in the long run, still prevents it from ignoring the random vector.

3.4. Quality and Diversity Metrics

When dealing with multiple plausible motions, or in general diverse solutions to a problem, evaluation is a challenge. The standard metrics used for deterministic motion prediction models are ill-suited to this task, because they typically compare the predictions to the ground truth, thus inherently penalizing diversity. For multiple motions, two

aspects are important: the *diversity* and the *quality*, or realism, of each individual motion. Prior work typically evaluates these aspects via human judgement. While human evaluation is highly valuable, and we will also report human results, it is very costly and time-consuming. Here, we therefore introduce two metrics that facilitate the quantitative evaluation of both quality and diversity of generated human motions. We additionally extend the Inception-Score [34] to our task.

To measure the quality of generated motions, we propose to rely on a binary classifier trained to discriminate real (ground-truth) samples from fake (generated) ones. The accuracy of this classifier on the test set is thus inversely proportional to the quality of the generated motions. In other words, high-quality motions are those that are not distinguishable from real ones. Note that we do not rely on adversarial training, i.e., we do not define a loss based on this classifier when training our model. To measure the diversity of the generated motions, a naive approach would consist of relying on the distance between the generated motion and a reference one. However, generating identical motions that are all far from the reference one would therefore yield a high value, while not reflecting diversity. To prevent this, we propose to make use of the average distance between all pairs of generated motions. A similar idea has been investigated to measure the diversity of solutions in other domains [43, 42].

The quality and diversity metrics can reliably evaluate a stochastic motion prediction model. While providing valuable information, drawing conclusion about the performance of a model is always easier with a single measure. To this end, we extend the Inception-Score (IS) [34] used to measure the quality of images produced by a generative model. Our extension to IS is twofold: **(1)** Inspired by [16], we extend IS to the conditional case, where the condition provides the core signal to generate the sample; **(2)** Our extended IS measures the quality and diversity of *sequential* solutions. To this end, we first train a strong skeleton-based action classifier [25] on ground-truth motions. With then compute the IS of each of the multiple motions generated for a given condition (observed motion), and report the mean IS and its standard deviation over all conditions. The reason behind reporting the mean IS over all conditions is to evaluating the diversity of generated motions given each observation. Note that studying IS only makes it hard to evaluate quality and diversity separately, and thus we still believe that all three metrics are required. Importantly, we show empirically that our proposed metrics are in line with human judgement, at considerably lower cost.

4. Experiments

We now evaluate the effectiveness of our approach at generating multiple plausible motions. To this end, we use Human3.6M [17] and the CMU Mocap dataset[1], two large publicly available motion capture datasets. In this section, we introduce the baselines and give information about the implementation details and evaluation metrics. We then provide all the experimental results.

Baselines. We compare our Mix-and-Match approach with the different means of imposing variation in motion prediction discussed in Section 2, i.e., concatenating the hidden state to a learned latent variable, Yan et al., [40], concatenating the pose to a learned latent variable at each time-step, Walker et al., [37], and adding a (transformed) random noise to the hidden state, Barsoum et al., [5]. For the comparison to be fair, we use 16 frames (i.e., 640ms) as observation to generate the next 60 frames (i.e., 2.4sec) for all baselines. All models are trained with the same motion representation, annealing strategy, backbone network, and losses, except for Barsoum et al., [5] which cannot make use of $\mathcal{L}_{prior}$.

Implementation Details. The motion encoders and decoders in our model are single layer GRU [9] networks, comprising 1024 hidden units each. For the decoders, we use a teacher forcing technique [39] to decode motion. At each time-step, the network chooses with probability P_{tf} whether to use its own output at the previous time-step or the ground-truth pose as input. We initialize $P_{tf} = 1$, and decrease it linearly at each training epoch such that, after a certain number of epochs, the model becomes completely autoregressive, i.e., uses only its own output as input to the next time-step. We train our model on a single GPU with the Adam optimizer [19] for 100K iterations. We use a learning rate of 0.001 and a mini-batch size of 64. To avoid exploding gradients, we use the gradient-clipping technique of [29] for all layers in the network. We implemented our model using the Pytorch framework of [30].

Evaluation Metrics. In addition to the metrics discussed in Section 3.4, we also report the standard ELBO metric (approximated by the reconstruction loss and the KL on the test set) and the sampling loss (S-MSE) of our approach and the state-of-the-art stochastic motion prediction techniques. However, evaluating only against one ground-truth motion (i.e., one sample from multi-modal distribution), as in MSE or S-MSE, can lead to a high score for one sample while penalizing other plausible modes. This behavior is undesirable since it cannot differentiate a multi-modal solution from a good, but uni-modal one. Similarly, the metrics in [40] or the approximate ELBO only evaluate quality given one single ground truth. While the ground truth has high quality, there exist multiple high quality continuations of an observation, which our proposed metric accounts for. As discussed in Section 3.4, we evaluate the quality and diversity of the predicted motions. Note, these metrics should be considered together, since each one taken separately does not provide a complete picture of how well a model can

[1] Available at `http://mocap.cs.cmu.edu/`.

Quantitative results on Human3.6M dataset

Method	ELBO ↓ (KL ↑)	Diversity ↑	Quality ↑	IS ↑	Tr KL ↑
Yan et al., [40]	0.51 (0.06)	0.26	0.45	1.9±0.4	0.08
Walker et al., [37]	2.08 (N/A)	1.70	0.13	1.8±0.6	N/A
Barsoum et al., [5]	0.61 (N/A)	0.48	0.47	2.1±1.3	N/A
`Mix-and-Match`	0.55 (2.03)	3.52	0.42	7.3±1.4	1.98

Quantitative results on CMU Mocap dataset

Method	ELBO ↓ (KL ↑)	Diversity ↑	Quality ↑	IS ↑	Tr KL ↑
Yan et al., [40]	0.25 (0.08)	0.41	0.46	2.4±0.1	0.01
Walker et al., [37]	1.93 (N/A)	3.00	0.18	1.4±0.4	N/A
Barsoum et al., [5]	0.24 (N/A)	0.43	0.45	2.0±1.0	N/A
`Mix-and-Match`	0.25 (2.92)	2.63	0.46	9.0±1.7	2.20

Table 1. Comparison of our approach with the stochastic motion prediction baselines on Human3.6M dataset (left) and CMU Mocap dataset (right). Tr KL stands for KL term at training convergence.

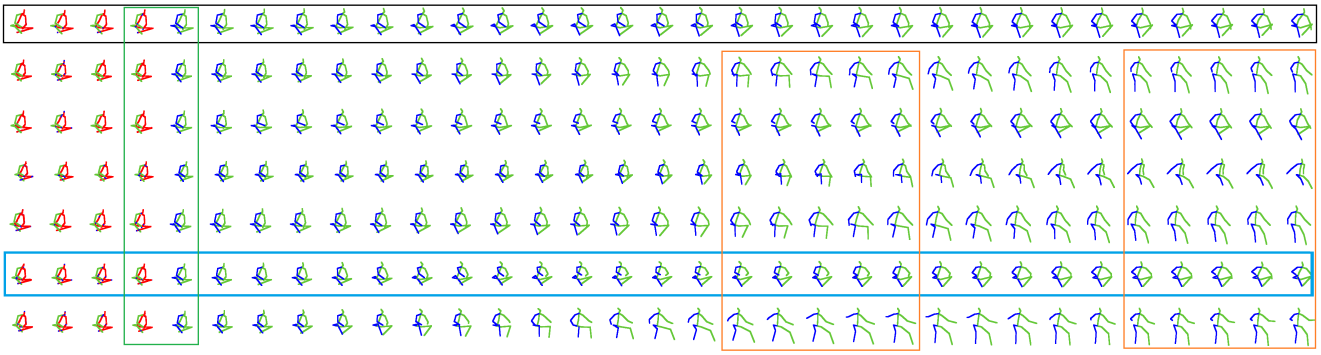

Figure 4. Qualitative evaluation of diversity. The first row (black box) shows the ground-truth motion. The next six rows depict six randomly generated motions (not cherry-picked) given the same observations (the first four poses of each motion). The green box shows the last observed frame and the first generated one, illustrating the consistency of the generated motions. The orange boxes show the diversity of the generated motions in different temporal windows. The blue box shows a randomly sampled motion whose poses are similar to the ground-truth ones. Best seen in color and zoomed in.

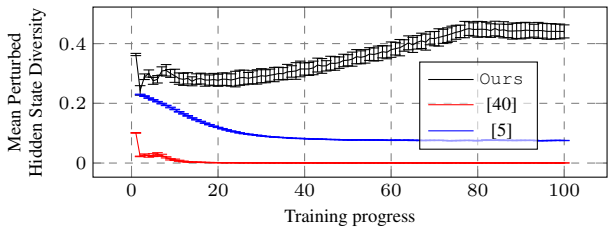

Figure 5. Diversity of K RNN decoder inputs, generated with $K = 50$ different random vectors. We report the mean diversity over $N = 50$ samples and the corresponding standard deviation.

predict *multiple plausible* future motions. For example, a model can generate diverse but unnatural motions, or, conversely, realistic but identical motions. To evaluate quality, as discussed in Section 3.4, we use a recurrent binary classifier whose task is to determine whether a sample comes from the ground-truth data or was generated by the model. We train such a classifier for each method, using 25K samples generated at different training steps together with 25K real samples, forming a binary dataset of 50K motions for each method. To evaluate diversity, as discussed in Section 3.4, we compute the mean Euclidean distance from each motion to all other $K - 1$ motions when generating $K = 50$ motions. To compute IS, we trained an action classifier [25] with 50K real motions. We then compute the IS for $K = 50$ samples per condition for 50 different conditions. We followed Section 3.4 to report IS. Furthermore, we also performed a human evaluation to measure the quality of the motions generated by each method. To this end, we asked eight users to rate the quality of 50 motions generated by each method, for a total of 200 motions. The ratings were defined on a scale of 1-5, 1 representing a low-quality motion and 5 a high-quality, realistic one. We then scaled the values to the range 0-50 to make them comparable with those of the binary classifier.

4.1. Comparison to the State-of-the-Art

In this section, we quantitatively compare our approach to the state-of-the-art stochastic motion prediction techniques in terms of approximate ELBO, Diversity, Quality, and IS on a held-out test set, as well as the training KL term at convergence. Table 1 shows the results on the Human3.6M and CMU Mocap datasets.

These results show that Mix-and-Match is highly capable of learning the variation in human motion while maintaining a good motion quality. This is shown by IS, Diversity, and Quality metrics, which should be considered together. It is also evidenced by the low reconstruction loss and higher KL term on the test set. The training KL term at convergence also shows that, in Mix-and-Match, the posterior does not collapse to the prior distribution, i.e., the model does not ignore the latent variable. While the MSE of our approach is slightly higher than that of Yan et al., [40] on Human3.6M and Barsoum et al., [5] on the CMU Mocap dataset, we effectively exploit the latent variables, as demonstrated by the KL term on the test set, the IS and diversity metric and the qualitative results provided in Fig. 4 and in the supplementary material. As evidenced by the examples of diverse motions generated by our model in Fig. 4, given a single observation, Mix-and-Match is able to gener-

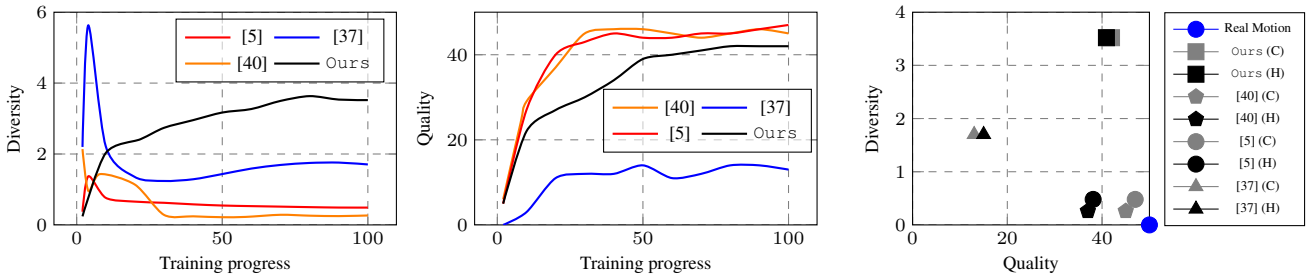

Figure 6. (Left) Diversity of our approach and the stochastic baselines. (Middle) Quality of our approach and the stochastic baselines. (Right) Comparing classifier-based and human evaluation of quality for our approach and the baselines, where the statistics correspond to evaluation after the models are fully trained. The numbers are provided in the supplementary material to facilitate future comparisons.

ate diverse, but natural motions[2].

4.2. Analysis on Diversity and Quality

To provide a deeper understanding of our approach, we evaluate different aspects of Mix-and-Match. All these experiments were done on Human3.6M. In the following, we first analyze the diversity in the hidden state space, i.e., the first part of the model where variation is imposed. We then evaluate the quality and diversity of prediction when tested at different stages of the training. We also perform a human evaluation on the quality of the generated motions, comparing it with our inexpensive, automatic quality metric. Finally, we compare Mix-and-Match with other stochastic techniques in terms of sampling error (S-MSE), i.e., by computing the error of the best of K generated motions given the ground-truth one. More experiments and visualizations are provided in the supplementary material.

Diversity in Hidden State Space. In Fig. 5, we plot the diversity of the representations used as input to the RNN decoders of [40] and [5], two state-of-the-art methods that are closest in spirit to our approach. Here, diversity is measured as the average pairwise distance across the $K = 50$ representations produced for a single series of observations. We report the mean diversity over 50 samples and the corresponding standard deviation. As can be seen from the figure, the diversity of [40] and [5] decreases as training progresses, thus supporting our observation that these models learn to ignore the perturbations. As evidenced by the black curve, which shows an increasing diversity as training progresses, our approach produces not only high-quality predictions but also truly diverse ones. The gradual but steady increase in diversity of our approach is due to our curriculum learning strategy described in Section 3.3. Without it, training is less stable, with large diversity variations.

Diversity and Quality in Motion Space. Now, we thoroughly compare our approach with state-of-the-art stochastic motion prediction models in terms of quality and diversity. The results of the metrics of Section 3.4 are provided in Fig. 6(Left and Middle) and those of the human evaluation in Fig. 6(Right). Below, we analyze the results of the different models.

As can be seen from Fig. 6, [40] tends to ignore the random variable z, thus ignoring the root of variation. As a consequence, it achieves a low diversity, much lower than ours, but produces samples of high quality, albeit almost identical, which is also shown in qualitatively in Fig. 3 of the supplementary material. We empirically observed that the magnitude of the weights acting on z to be orders of magnitude smaller than that of acting on the condition, 0.008 versus 232.85 respectively. Note that this decrease in diversity occurs after 16K iterations, indicating that the model takes time to identify the part of the hidden state that contains the randomness. Nevertheless, at iteration 16K, prediction quality is low, and thus one could not simply stop training at this stage. Note that the lack of diversity of [40] is also evidenced by Fig. 5. As can be verified in Fig. 6(Right), where [40] appears in a region of high quality but low diversity, the results of human evaluation match those of our classifier-based quality metric.

Fig. 6 also evidences the limited diversity of the motions produced by [5] despite its use of random noise during inference. Note that the authors of [5] mentioned in their paper that the random noise was added to the hidden state. Only by studying their publicly available code[3] did we understand the precise way this combination was done. In fact, the addition relies on a parametric, linear transformation of the noise vector. That is, the perturbed hidden state is obtained as $h_{perturbed} = h_{original} + W^{z \to h} z$. Because the parameters $W^{z \to h}$ are *learned*, the model has the flexibility to ignore z (the magnitude of $W^{z \to h}$ is in the order of $O(1e^{-3})$), which causes the behavior observed in Figs. 6 and 5. Note that the authors of [5] acknowledged that, despite their best efforts, they noticed very little variations between predictions obtained with different z values. By depicting [5] in a region of high quality but low diversity, the human evaluation results in Fig. 6(Right) again match those of our classifier-based quality metric.

As can be seen in Fig. 6(Left and Middle), [37] pro-

[2] See the video of our results in the supplementary material.

[3] https://github.com/ebarsoum/hpgan

Walking						
Method	80ms	160ms	320ms	400ms	560ms	1000ms
Yan et al., [40]	0.73	0.79	0.90	0.93	0.95	1.05
Barsoum et al., [5]	0.61	0.62	0.71	0.79	0.83	1.07
Walker et al., [37]	0.56	0.66	0.98	1.05	1.28	1.60
Mix-and-Match	0.33	0.48	0.56	0.58	0.64	0.68
Smoking						
Method	80ms	160ms	320ms	400ms	560ms	1000ms
Yan et al., [40]	1.00	1.14	1.43	1.44	1.68	1.99
Barsoum et al., [5]	0.64	0.78	1.05	1.12	1.64	1.84
Walker et al., [37]	0.59	0.83	1.25	1.36	1.67	2.03
Mix-and-Match	0.23	0.42	0.79	0.77	0.82	1.25

Eating						
Method	80ms	160ms	320ms	400ms	560ms	1000ms
Yan et al., [40]	0.68	0.74	0.95	1.00	1.03	1.38
Barsoum et al., [5]	0.53	0.67	0.79	0.88	0.97	1.12
Walker et al., [37]	0.44	0.60	0.71	0.84	1.05	1.54
Mix-and-Match	0.23	0.34	0.41	0.50	0.61	0.91
Discussion						
Method	80ms	160ms	320ms	400ms	560ms	1000ms
Yan et al., [40]	0.80	1.01	1.22	1.35	1.56	1.69
Barsoum et al., [5]	0.79	1.00	1.12	1.29	1.43	1.71
Walker et al., [37]	0.73	1.10	1.33	1.34	1.45	1.85
Mix-and-Match	0.25	0.60	0.83	0.89	1.12	1.30

Table 2. Quantitative comparison of the S-MSE against stochastic baselines for four actions of the Human3.6M dataset.

Walking						
Method	80ms	160ms	320ms	400ms	560ms	1000ms
Zero Velocity	0.39	0.86	0.99	1.15	1.35	1.32
AGED [14]	0.22	0.36	0.55	0.67	0.78	0.91
Imitation [38]	0.21	0.34	0.53	0.59	0.67	0.69
LSTM-3LR [12]	1.18	1.50	1.67	1.76	1.81	2.20
SRNN [18]	1.08	1.34	1.60	1.80	1.90	2.13
DAE-LSTM [13]	1.00	1.11	1.39	1.48	1.55	1.39
GRU [28]	0.28	0.49	0.72	0.81	0.93	1.03
LTD [27]	0.18	0.31	0.49	0.56	0.65	0.67
Mix-and-Match	0.33	0.48	0.56	0.58	0.64	0.68
Smoking						
Method	80ms	160ms	320ms	400ms	560ms	1000ms
Zero Velocity	0.26	0.48	0.97	0.95	1.02	1.69
AGED [14]	0.27	0.43	0.82	0.84	1.06	1.21
Imitation [38]	0.23	0.44	0.86	0.85	0.95	1.63
LSTM-3LR [12]	2.05	2.34	3.10	3.18	3.24	3.42
SRNN [18]	1.90	2.30	2.90	3.10	3.21	3.23
DAE-LSTM [13]	0.92	1.03	1.15	1.25	1.38	1.77
GRU [28]	0.33	0.61	1.05	1.15	1.25	1.50
LTD [27]	0.22	0.41	0.86	0.80	0.87	1.57
Mix-and-Match	0.23	0.42	0.79	0.77	0.82	1.25

Eating						
Method	80ms	160ms	320ms	400ms	560ms	1000ms
Zero Velocity	0.27	0.48	0.73	0.86	1.04	1.38
AGED [14]	0.17	0.28	0.51	0.64	0.86	0.93
Imitation [38]	0.17	0.30	0.52	0.65	0.79	1.13
LSTM-3LR [12]	1.36	1.79	2.29	2.42	2.49	2.82
SRNN [18]	1.35	1.71	2.12	2.21	2.28	2.58
DAE-LSTM [13]	1.31	1.49	1.86	1.89	1.76	2.01
GRU [28]	0.23	0.39	0.62	0.76	0.95	1.08
LTD [27]	0.16	0.29	0.50	0.62	0.76	1.12
Mix-and-Match	0.23	0.34	0.41	0.50	0.61	0.91
Discussion						
Method	80ms	160ms	320ms	400ms	560ms	1000ms
Zero Velocity	0.31	0.67	0.94	1.04	1.41	1.96
AGED [14]	0.27	0.56	0.76	0.83	1.25	1.30
Imitation [38]	0.27	0.56	0.82	0.91	1.34	1.81
LSTM-3LR [12]	2.25	2.33	2.45	2.46	2.48	2.93
SRNN [18]	1.67	2.03	2.20	2.31	2.39	2.43
DAE-LSTM [13]	1.11	1.20	1.38	1.42	1.53	1.73
GRU [28]	0.31	0.68	1.01	1.09	1.43	1.69
LTD [27]	0.20	0.51	0.77	0.85	1.33	1.70
Mix-and-Match	0.25	0.60	0.83	0.89	1.12	1.30

Table 3. Comparison against deterministic motion prediction techniques for four actions of the Human3.6M dataset.

duces motions with higher diversity than [5, 40], but of much lower quality. The main reason behind this is that the random vectors that are concatenated to the poses at each time-step are sampled independently of each other, which translates to discontinuities in the generated motions. Human evaluation in Fig.6(Right) further confirms that [37]'s results lie in a low-quality, medium-diversity region.

The success of our approach is confirmed by Fig. 6(Left and Middle). Our model generates diverse motions, even after a long training time, and the quality of these motions is high. While this quality is slightly lower than that of [5, 40] when looking at our classifier-based metric, it is rated higher by IS and humans, as can be verified from Fig. 6(Right) and Table 1. Altogether, these results confirm the ability of our approach to generate highly diverse yet realistic motions.

Evaluating the Sampling Error. We now quantitatively compare our approach with other stochastic baselines in terms of sampling error (aka S-MSE). To this end, we follow the evaluation setting of deterministic motion prediction (as in [12, 31, 32, 28, 14]) which allows further comparisons to deterministic baselines. We report the standard metric, i.e., the Euclidean distance between the generated and ground-truth Euler angles (aka MAE). To evaluate this metric for our method and the stochastic motion prediction models, which generate multiple, diverse predictions, we make use of the best sample among the K generated ones with $K = 50$ for the stochastic baselines and for our approach. This evaluation procedure aims to show that, among the K generated motions, at least one is close to the ground truth. As shown in Table 2, by providing higher diversity, our approach outperforms the baselines. Similarly, in Table 3, we compare the best of $K = 50$ sampled motions for our approach with the deterministic motion prediction techniques. Note that the goal of this experiment is not to provide a fair comparison to deterministic models, but to show that, among the diverse set of motions generated by our model, there exists at least one motion that is very close to the ground-truth one. The point of bringing the MAE of other deterministic methods, is to show how good deterministic models, with sophisticated architectures and complicated loss functions, perform on this task.

5. Conclusion

In this paper, we have proposed an effective way of perturbing the hidden state of an RNN such that it becomes capable of learning the multiple modes of human motions. Our evaluation of quality and diversity, based on both new quantitative metrics and human judgment, have evidenced that our approach outperforms existing stochastic methods. Generating diverse plausible motions given limited observations has many applications, especially when the motions are generated in an action-agnostic manner, as done here. For instance, our model can be used for human action forecasting [33, 2, 35, 1, 3], where one seeks to anticipate the action as early as possible, or for motion inpainting, where, given partial observations, one aims to generate multiple in-between solutions. In the future, we will therefore investigate the use of our approach in such applications.

References

[1] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Encouraging lstms to anticipate actions very early. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[2] Mohammad Sadegh Aliakbarian, Fatemehsadat Saleh, Basura Fernando, Mathieu Salzmann, Lars Petersson, and Lars Andersson. Deep action-and context-aware sequence learning for activity recognition and anticipation. *arXiv preprint arXiv:1611.05520*, 2016.

[3] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Viena^2: A driving anticipation dataset. In *Asian Conference on Computer Vision*, pages 449–466. Springer, 2018.

[4] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2745–2754, 2017.

[5] Emad Barsoum, John Kender, and Zicheng Liu. Hp-gan: Probabilistic 3d human motion prediction via gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1418–1427, 2018.

[6] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[7] Judith Butepage, Michael J Black, Danica Kragic, and Hedvig Kjellstrom. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6158–6166, 2017.

[8] Judith Bütepage, Hedvig Kjellström, and Danica Kragic. Anticipating many futures: Online human motion prediction and generation for human-robot interaction. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[10] Jesse Engel, Matthew Hoffman, and Adam Roberts. Latent constraints: Learning to generate conditionally from unconditional generative models. *arXiv preprint arXiv:1711.05772*, 2017.

[11] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018.

[12] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[13] Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pages 458–466. IEEE, 2017.

[14] Liang-Yan Gui, Yu-Xiong Wang, Xiaodan Liang, and José MF Moura. Adversarial geometry-aware human motion prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 786–803, 2018.

[15] Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and José MF Moura. Few-shot human motion prediction via meta-learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 432–450, 2018.

[16] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

[17] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[18] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5308–5317, 2016.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[21] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015.

[22] Jogendra Nath Kundu, Maharshi Gor, and R Venkatesh Babu. Bihmp-gan: Bidirectional 3d human motion prediction gan. *arXiv preprint arXiv:1812.02591*, 2018.

[23] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[24] Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5226–5234, 2018.

[25] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *International Joint Conferences on Artificial Intelligence*, 2018.

[26] Xiao Lin and Mohamed R Amer. Human motion modeling using dvgans. *arXiv preprint arXiv:1804.10652*, 2018.

[27] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. *arXiv preprint arXiv:1908.05436*, 2019.

[28] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks.

In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4674–4683. IEEE, 2017.

[29] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318, 2013.

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[31] Dario Pavllo, Christoph Feichtenhofer, Michael Auli, and David Grangier. Modeling human motion with quaternion-based neural networks. *arXiv preprint arXiv:1901.07677*, 2019.

[32] Dario Pavllo, David Grangier, and Michael Auli. Quaternet: A quaternion-based recurrent model for human motion. *arXiv preprint arXiv:1805.06485*, 2018.

[33] Cristian Rodriguez, Basura Fernando, and Hongdong Li. Action anticipation by predicting future dynamic images. In *European Conference on Computer Vision*, pages 89–105. Springer, 2018.

[34] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[35] Yuge Shi, Basura Fernando, and Richard Hartley. Action anticipation with rbf kernelized feature mapping rnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 301–317, 2018.

[36] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[37] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 3352–3361. IEEE, 2017.

[38] Borui Wang, Ehsan Adeli, Hsu kuang Chiu, De-An Huang, and Juan Carlos Niebles. Imitation learning for human pose prediction. *arXiv preprint arXiv:1909.03449*, 2019.

[39] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[40] Xinchen Yan, Akash Rastogi, Ruben Villegas, Kalyan Sunkavalli, Eli Shechtman, Sunil Hadap, Ersin Yumer, and Honglak Lee. Mt-vae: Learning motion transformations to generate multimodal human dynamics. In *European Conference on Computer Vision*, pages 276–293. Springer, 2018.

[41] Xinchen Yan, Jimei Yang, Kihyuk Sohn, and Honglak Lee. Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision*, pages 776–791. Springer, 2016.

[42] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tiangchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[43] Ye Yuan and Kris Kitani. Diverse trajectory forecasting with determinantal point processes. *arXiv preprint arXiv:1907.04967*, 2019.