A Scalable Approach to Predict Multi-Agent Motion for Human-Robot Collaboration

Mohammad Samin Yasar and Tariq Iqbal

***Abstract*—Human motion prediction is considered a key component for enabling fluent human-robot collaboration. The ability to anticipate the motion and subsequent intent of the partner(s) remains a challenging task due to the complex and interpersonal nature of human behavior. In this work, we propose a novel sequence learning approach that learns a robust representation over the observed human motion and can condition future predictions over a subset of past sequences. Our approach works for both single and multi-agent settings and relies on an interpretable latent space that has the implicit benefit of improving human motion understanding. We evaluated the proposed approach by comparing its performance against state-of-the-art motion prediction methods on single, multi-agent, and human-robot collaboration datasets. The results suggest that our approach outperforms other methods over all the evaluated temporal horizons, for single-agent and multi-agent motion prediction. The improved performance of our approach for both single and multi-agent settings, coupled with an interpretable latent space, can enable close-proximity human-robot collaboration.**

***Index Terms*—Human detection and tracking, human-robot collaboration, intention recognition.**

I. Introduction

UNDERSTANDING human motion is a crucial skill for robots to coexist and collaborate with humans [1]–[3]. Humans develop the ability to engage in joint action during infancy and early childhood, through a combination of observation, active participation and explicit teaching [4], [5]. As such, humans are innately adept at anticipating the motion and intent of other persons over varying horizons [6], [7]. This is best observed in team activities, where two or more individuals can understand and predict each other's motion [5], [8]. Along these lines, for robots to fluently collaborate with humans, they need to combine aspects of perception, representation, and motion analysis, to accurately anticipate the motion of surrounding individuals [9]–[13]. In addition, the robot's perception and decision-making processes need to be explainable to human collaborators for enabling close human-robot collaboration.

Human motion is often modeled by tracking the movement of the skeletal joints over time [14]–[18]. Several approaches have modeled the problem of predicting human motion as that of forecasting future trajectories, conditioned on past observed trajectories, in a sequence-to-sequence manner [15]–[17]. Prior work can be broadly categorized into deterministic approaches: learning point estimates over future trajectory [15]–[17], and probabilistic approaches: learning a distribution over future trajectories using latent variables [4], [18], [19]. Although the aforementioned works have shown promising results, predicting human motion *quantitatively*, in terms of some evaluation metric such as Mean Squared Error and *qualitatively* (see Fig. 1), in terms of generating feasible and realistic motion remains a challenging task [16], [17]. This highlights the need to learn a more robust representation of observed trajectories.

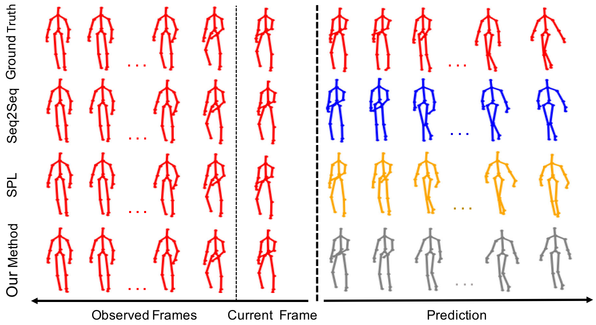

Fig. 1. Qualitative Performance of different motion prediction methods for *walking* on UTD-MHAD. Our method produces more feasible joint poses by maintaining relative orientation of each joints, while achieving the best quantitative performance.

Predicting the motion of just one person is not enough for a robot to be successful in a team. It is expected to work with multiple people and needs to capture the inter-agent dynamics to accurately predict the motion of all individuals. Prior work on multi-agent forecasting has primarily used the global motion (2D positions) for modeling the interaction among all the agents [20]–[23]. These approaches do not consider the local pose or skeletal joints of the humans, thus only modeling coarse information about human trajectory. To incorporate skeletal pose, recent work introduced a joint-learning framework that models both skeleton positions and global 2D positions, for multi-agent settings [24]. However, the approach relies on pooling mechanisms (e.g., [20], [21]) to model the interaction among multiple agents, which are prone to losing valuable information while being invariant to small changes in input [25], [26].

To address the above challenges, we propose an encoder-decoder approach that is *scalable:* predicting human motion for single and multiple agents, and *interpretable:* disentangling relevant aspects of human motion. The encoder architecture of our approach differs from prior works [15], [16], [21], [27] by

Manuscript received October 15, 2020; accepted January 21, 2021. Date of publication February 11, 2021; date of current version March 4, 2021. This letter was recommended for publication by Associate Editor X. Zhong and Editor J. Yi upon evaluation of the reviewers' comments. This work was supported by the CCAM Innovation Award. The authors thank Tim Bakker, Tomonari Furukawa, and Qing Chang for their support. *(Corresponding author: Mohammad Yasar.)*

The authors are with the School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia 22903 USA (e-mail: msy9an@virginia.edu; tiqbal@virginia.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3058917, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3058917

explicitly considering velocity, and acceleration features in addition to skeletal positions, to obtain a more salient representation over past motion. These features are fed to an attention mechanism [28], which learns to adaptively weigh the different motion features and is more robust at capturing relevant information, compared to pooling mechanisms. The output from the attention mechanism is used to obtain the latent representation, which comprises of continuous and categorical random variables. The decoder then uses these latent variables to forecast future trajectories in an auto-regressive manner. We differ from previous work by learning to condition the decoder output on a subset of the past sequences, instead of just the last predicted frame.

In settings of more than one agent, we use separate encoders and decoders to model the motion for each agent. To model inter-agent dynamics our approach relies on a novel attention-based mechanism that learns to weigh relevant features from each agent to produce a disentangled multi-agent representation. This is used to compute the shared latent representation for all agents, which models the categorical and continuous aspects of multi-agent interaction. The output of the latent space is then used by each agent-specific decoder to condition its prediction between the immediate agent-specific prediction and the latent variables that represent multi-agent interaction.

We evaluated the performance of our approach on single-agent settings on the UTD-MHAD [29], multi-agent settings on the NTU RGB+D 60 [30] and CMU Panoptic [31] datasets, and human-robot collaboration scenarios on the KTH Human-Robot Collaboration (KTH-HRC) dataset [4]. The results suggest that our approach outperformed state-of-the-art human motion prediction methods over all the evaluated horizons for single-agent and multi-agent settings. Finally, we provide an interpretation of the underlying generative process of human motion by exploring the latent space. Our findings suggest that the categorical latent variables learn to segment an action into separate action primitives while the continuous latent variables learn to cluster activities with similar spatial semantics.

II. Related Work

Human motion prediction: Recent work on human motion prediction has predominantly posed the problem as that of sequence learning, modeled using Recurrent Neural Nets in an encoder-decoder framework [14]–[18]. Martinez *et al.* [16] showed that weight sharing between the encoder and decoder results in quicker convergence. Furthermore, they model velocity representation at the decoder by introducing a residual connection. To explicitly encode the skeletal hierarchy, prior work has modeled the kinematics chain at the encoder by dividing the skeleton into 5 major clusters [14] or following the kinematic chain starting from the end-effectors [32]. Aksan *et al.* [17] proposed structured prediction at the decoder, by introducing a Structured Prediction Layer which decomposes the model prediction into individual skeletal joints, each predicted in a hierarchical sequence. While most works on motion prediction adopt a deterministic approach, recent work has approached the problem as that of learning a probability density function of future human poses conditioned on previous poses [4], [18], [19], [27].

Butepage *et al.* [18] and Toyer *et al.* [19] adopted the Variational Autoencoder (VAE) framework for motion prediction, which rely on learning a functional mapping from the data space to the latent space at the encoder, with the decoder sampling from this latent space to generate future human motion. Barsoum *et al.* [27] proposed a modified version of Wasserstein GAN (WGAN-GP) with the model input being a sequence of past human poses plus a random vector z.

Multi-agent motion prediction: Multi-agent forecasting is widely considered a challenging problem as the agents' policies are not directly accessible. Several data-driven approaches have been applied to forecast complex interactions in social navigation [20], [21], [33], autonomous vehicles [22], [23], [34] and HRI settings [35], [36]. Alahi *et al.* [20] introduced social-LSTM, which uses agent-specific LSTMs to summarize past observations of each agent. The hidden states of the neighboring LSTMs are connected through a social pooling strategy and used as the input to the LSTM cell at the next timestep. Gupta *et al.* [21] proposed Social GAN, which introduced a computationally efficient pooling mechanism comprising of a Multi-Layer Perceptron followed by max pooling. While the aforementioned works only consider the global motion of the agents, in particular 2D locations, Adeli *et al.* [24] jointly modeled global and local movement by incorporating skeleton positions. Despite the promising performances of these methods, the pooling mechanism commonly used in these approaches runs the risk of losing valuable information while being invariant to small changes in input [25], thereby learning a sub-optimal representation.

Human motion interpretation: Interpreting the learned representation of deep learning frameworks is crucial to their acceptability for any application. The problem of latent space learning and interpretation for images has been extensively studied and introduced several seminal approaches [37]–[40]. In comparison, work on understanding the underlying generative process of human motion is less explored. For human-robot collaboration, robot perception needs to be explainable. Prior work has modeled various aspects of human-robot collaboration from human motion [18] to robot motion [41] and emotion [42] using continuous latent variables, while providing an intuitive explanation of the learned latent representation. However, these approaches learn the latent representations over simple motion (*reaching* or *pouring*) and cannot not capture the high level dynamics of human motion.

Although the aforementioned works show promising results, learning effective representations that summarize the observed trajectory at the encoder remains an open problem. In addition, the decoder network in most approaches condition only on the past generated frame. This results in performance degradation over long-term horizons and is not suited for multi-agent settings where there is a need to consider cross-agent interaction. To this end, prior approaches rely on pooling over encoder representations of multiple agents, which can lead to losing relevant information. Finally, prior works on human motion interpretation focus on learning representations over simple actions and fail to capture the high level dynamics of human motion. To address these challenges, we propose an encoder-decoder approach for human motion prediction, which we describe in Section IV.

III. Problem Formulation

Our goal is to accurately predict the motion of all agents in a given workspace. We assume that the number of agents, m is known. In all our formulations, we use superscript to represent agents and subscript to represent time.

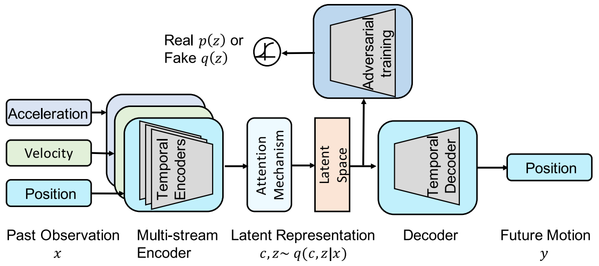

Fig. 2. Proposed framework for single-agent setting.

For simplicity, let us first assume that there is one agent a and we have access to the agent's trajectory, spanning time $t = 1$ to τ, with observed trajectory frames: $\mathbf{X}^a = \{x_1^a, \ldots, x_\tau^a\}$. We pose the motion prediction problem as predicting future trajectory frames over a horizon H: $\mathbf{Y}^a = \{y_{\tau+1}^a, \ldots, y_{\tau+H}^a\}$, conditioned on the observed frames $\mathbf{X}^a$. Each frame $x_t^a \in \mathbb{R}^N$ denotes the N-dimensional body pose. N depends on the number of joints in the skeleton, J and the dimension of the joints D, where $N = J \times D$.

We assume that future human pose is conditioned on the past observed or generated poses, and predict each frame in an auto-regressive manner as formulated below:

$$p_\theta(\mathbf{Y}^a) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(y_\delta^a | y_{\tau:\delta-1}^a, x_{1:\tau}^a) \tag{1}$$

where the joint distribution is parameterized by θ.

In the case of multiple agents, we assume that the future pose for each agent is conditioned on the observed poses of all the agents and generated pose of the specific agent. As such, we can extend Eq. 1 for each agent a as follows:

$$p_\theta(\mathbf{Y}^a) = \prod_{\delta=\tau+1}^{\tau+H} p_\theta(y_\delta^a | y_{\tau:\delta-1}^a, x_{1:\tau}^{1:m}); \forall a = 1, \ldots, m \tag{2}$$

IV. Human Motion Prediction

Our approach has the overarching goal of accurately predicting human motion while being scalable and interpretable. It comprises of an encoder-decoder, trained end-to-end, with adversarial regularization on the latent variables. To address the challenges of learning a robust representation, the encoder explicitly models position, velocity, and acceleration information. The decoder conditions its output on both the latent representation and the immediate past frame, thus attaining performance gain over long horizons. For multi-agent settings, our approach uses an attention mechanism to model the inter-agent dynamics, thus learning a more robust representation. We will first describe our framework for single-agent motion prediction and then discuss its scalability for predicting the motion of multiple agents.

A. Single-Agent Motion Prediction

Our framework for a single-agent setting comprises of one encoder-decoder, along with adversarial training (see Fig. 2).

Multi-stream Encoder: The encoder aims to learn a spatio-temporal representation over the past observation for a given agent. To obtain a rich and more robust representation over the past trajectories, we extract the past velocity and acceleration features along with the provided positional values, thus forming a multi-stream input for the encoder. The velocity and acceleration features are first and second-order derivative of the position values for each skeleton joint.

As we pose this as a sequence learning problem, we employ Recurrent Neural Networks, in particular unidirectional Gated Recurrent Units (GRU), to extract temporal feature representations for each stream. Our choice of unidirectional GRUs over a bi-directional architecture is motivated by our need to predict human motion in real-time. We choose GRUs due to their comparative performance to LSTMs while having computational advantages. For each stream, the stream-specific GRU aims to encode the spatio-temporal information over the input sequence, which is formulated as:

$$h_{s,t} = GRU(h_{s,t-1}, x_{s,t}, \phi_s) \tag{3}$$

where s represents position, velocity, or acceleration. Here, $x_{s,t}$ represents the input to the GRU at time t and will take the value of $x_{pos,t}, x_{vel,t}, x_{acc,t}$ for position, velocity and acceleration, respectively. $h_{s,t-1}$ represents the past hidden output and ϕ_s represents the stream-specific encoder weights for the GRU. The output from each GRU is passed to a multi-head self-attention module [28]. The attention module is tasked to sparsely and adaptively extract the salient features from the three streams.

$$h_t = Concat(h_{pos,t}, h_{vel,t}, h_{acc,t});\ h_{att,t} = Att(h_t, \phi_{att}) \tag{4}$$

In the self-attention module the concatenated output, h_t is at first linearly projected to query (Q), key (K), and value (V) embedding for each head. The embeddings are used to compute attention weights using the scaled-dot product softmax (sf) approach. The overall functions for each head in the multi-head self-attention module are formulated below:

$$\begin{gathered} Q = h_t W^Q;\ K = h_t W^K; V = h_t W^V \\ Att(Q, K, V) = sf\left(\frac{QK^T}{\sqrt{d_k}}\right) V \end{gathered} \tag{5}$$

where, W^Q, W^K, W^V represent the linear projection weights and $\frac{1}{\sqrt{d_k}}$ is the scaling factor for calculating the attention weights.

Latent Variables: Our proposed approach aims to learn a distribution over past observations similar to previous work [4], [19], [32], but differs in terms of the latent space representation and regularization. The core assumption underlying such approaches is that the past observations and future trajectories are generated by some random process involving unobserved latent variables. Unlike prior approaches, our framework models both continuous Z and categorical random variables C as part of the latent space.

In line with prior work on representation learning for images [38], [39], our framework augments the continuous latent distribution with a relaxed discrete distribution, but for human motion modeling. The motivation here is to disentangle and model continuous aspects of human motion such as the style of the agent, as well as discrete information such as class activity or action primitive.

To obtain the continuous latent variable z_t, the output from the self-attention module is passed through a linear layer (Lin), whereas in the case of the categorical latent variable c_t, the output from the self-attention module is passed through a linear layer

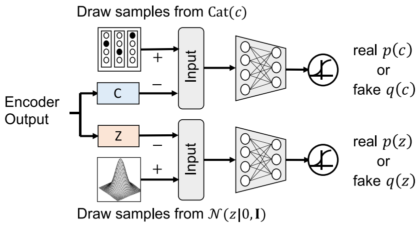

Fig. 3. Adversarial training over the latent space.

followed by a softmax (sf) layer.

$$z_t = Lin(h_{att,t});\ h_{c,t} = Lin(h_{att,t});\ c_t = sf(h_{c,t}) \qquad (6)$$

Discriminators: In line with previous frameworks on latent space learning and regularization, such as Variational Autoencoders (VAEs) [37], Joint-VAE [39] and Adversarial Autoencoders (AAE) [38], we enforce a prior on the latent variables. We differ from prior work on motion generation that use KL-divergence for enforcing a prior [19], [32], instead using adversarial training, thus adopting the AAE framework for motion generation [38]. Our choice of using adversarial training is to avoid tuning the KL-divergence loss that is often small compared to the reconstruction loss and requires a scaling factor β as well as an annealing schedule.

In our framework, the encoder aims to confuse the discriminators by trying to ensure that its output is similar to the aggregated prior. The discriminators are trained to distinguish the true samples generated using a given prior, from the latent space output of the encoder, thus establishing a min-max adversarial game between the networks [38], [40].

We use two discriminators, one for the continuous latent variable and the other for the categorical latent variable, as shown in Fig. 3. The discriminators compute the probability that a point z_t or c_t is a sample from the prior distribution that we are trying to model (positive samples), or from the latent space (negative sample). The discriminator loss, which is high if the generated sample from the encoder is coming from a different distribution compared to the prior, is used to update the parameters of the encoder, thus enforcing it to produce samples similar to the prior. We use a Gaussian prior for continuous latent variables and a uniform distribution prior for categorical latent variables.

Decoder: The decoder is auto-regressive, i.e., it uses the output of previous timesteps to predict the current pose, and has only one stream: position. The input to the decoder is the latent representation, summarizing the past observations as well as the immediate hidden representation of the last predicted frame. This is passed to a multi-head self-attention module, similar to one at the encoder, which learns the attention weights between the previous output and the latent variables that summarize past frames.

The first part of the decoder is a GRU cell, that takes as input the output of the multi-head self-attention module as well as the output of the last timestep. This is followed by either a fully connected layer or a Structured Prediction Layer (SPL) [17], which aim to explicitly model the spatial structure of the joints by hierarchically predicting each joint, instead of treating each joint individually. The operations at the decoder are formulated as follows:

$$\begin{aligned} p_t &= Concat(z_t, c_t, h_{dec,t-1});\ p_{att,t} = Att(p_t, \phi_{att}) \\ h_{dec,t} &= GRU(S_{t-1}, p_{att,t}, \phi_{pos});\ S_t = \gamma(h_{dec,t}) \end{aligned} \qquad (7)$$

where z_t and c_t are the latent variables, $h_{dec,t-1}$ is the previous hidden output of the GRU. $p_{att,t}$ is the output of the attention mechanism in the decoder, which is passed to the GRU along with the previous GRU output S_{t-1}. ϕ_{att} and ϕ_{pos} represents the weights of the attention module and GRU cell respectively. γ represents the output layer of the decoder with $\mathbf{S_t}$ being the *predicted motion at time* t. We add a residual connection between decoder output at the last and current timestep, which improves short-term prediction and result in smoother output sequence [16].

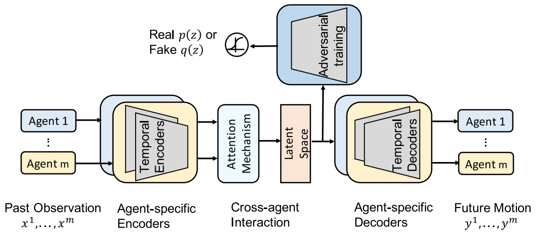

Fig. 4. Proposed framework for multi-agent setting.

B. Multi-Agent Motion Prediction

In addition to addressing the challenges of learning a robust representation and improving the interpretability of single-agent motion prediction, our approach can be scaled to predict motion for multiple agents.

Multi-stream Encoder: Each agent's motion is modeled by an agent-specific multi-stream encoder that learns a spatio-temporal representation over the past trajectories. The operations per-agent are similar to the ones in Eq. 3. For m agents, there will be m number of encoders and decoders, matching the number of agents (see Fig. 4).

To obtain a robust cross-agent interaction, the output of all the encoders is passed to a multi-head self-attention module. The operations can be summarized as:

$$\begin{aligned} h_t^a &= Concat(h_{pos,t}^a, h_{vel,t}^a, h_{acc,t}^a) \\ h_t &= Concat(h_t^a, \ldots, h_t^m); h_{att,t} = Att(h_t, \phi_{att}) \end{aligned} \qquad (8)$$

where h_t^a represents the agent-specific multi-stream output from each encoder. h_t is the concatenated representation for all agents and $h_{att,t}$ is the output of the attention module, representing the cross-agent interaction. We use the attention mechanism to disentangle and extract relevant multi-agent features from agent-specific representations while addressing the limitations of (max, average) pooling, which tend to summarize and thereby lose valuable information.

Latent variables and Discriminators: For multi-agent settings, we use the formulations of Eq. 6 to obtain the latent variables. Here, the latent variables represent the joint motion segment and the spatial semantics of all the agents. As the underlying functions for the discriminators and latent space remain unchanged, our approach is robust to the number of agents and can model interactions among all the agents.

Decoder: Each agent will have a specific decoder that auto-regressively predicts the motion for that agent only. The inputs to the decoder are the latent variables as well as the hidden representation of the last frame. This is passed to a self-attention module that learns the attention weights between the immediate agent-specific past output and the latent variables that represent multi-agent interaction. This allows the decoder to better capture inter-agent dynamics as well as condition its output on a subset of past frames.

The representation obtained from the self-attention module is fed to the GRU cell along with output of the last timestep. The output from the GRU is passed to a linear layer, with S_t^a being the predicted motion of agent a at time t. The operations at each decoder are formulated as follows:

$$p_t^a = Concat(z_t, c_t, h_{dec,t-1}^a);\; p_{att,t}^a = Att(p_t^a, \phi_{att}^a)$$
$$h_{dec,t}^a = GRU(S_{t-1}^a, p_{att,t}^a, \phi_{pos}^a);\; S_t^a = \gamma(h_{dec,t}^a) \tag{9}$$

V. Experimental Setup

A. Datasets

We evaluated the performance of our approach by applying it on three widely used human-activity and social interaction datasets: UTD-MHAD [29], NTU RGB+D 60 [30] and CMU Panoptic [31]. Furthermore, we evaluated our approach on the KTH Human-Robot Collaboration (KTH-HRC) dataset [4]. For single-agent motion prediction, we conducted experiments on the UTD-MHAD. The dataset contains 27 action classes covering activities from hand gestures to training exercises: providing a range of relevant and diverse activities for human-robot collaboration. We used skeleton data for predicting human motion, following previous work in this domain [14]–[18], and considered each of the 20 provided joints. We used the cross-subject evaluation scheme, training and validating on odd-numbered subjects while testing on even-numbered subjects.

For multi-agent motion prediction, we conducted experiments on the NTU-RGB+D 60 [30] and CMU Panoptic [31] datasets. For NTU-RGB+D 60 dataset, we focused on the action classes involving more than one agent, resulting in 11 joint actions in total, ranging from punching to hugging, similar to previous work [24]. We used the cross-subject evaluation scheme [30], with 20 subjects for training and validation and a separate 20 for testing. For the CMU Panoptic dataset, we focused on the Haggling action, which consisted of more than two agents and had a defined training and testing protocol. Similar to the single-agent setup, we used the skeleton modality and all provided joints of each agent for motion prediction across all methods. While having access to a different modality, such as RGB data, can potentially improve model performance, prior work has shown that the improvement is only marginal due to the constrained environmental setup in which the data were collected [24].

Finally, the KTH-HRC dataset [4] comprised 4 human-robot collaboration actions ranging from handshaking to hand wave. We used two experimental setups. First, we train our model on Human-Robot Collaboration (HRC) data and set aside the last 20% of all the trials for testing, in keeping with [4]. Second, we train the model on Human-Human Collaboration (HHC) data and test on HRC data. We used the same four joint positions as the original paper [4].

B. State-of-The-Art Methods and Baselines

For evaluating our model on single-agent settings, we compared against two state-of-the-art approaches: Seq2Seq-sampling [16], Seq2Seq-sampling-SPL [17], and the zero-velocity baseline [16]. The Seq2Seq-sampling approach is based on the sequence-to-sequence learning framework but introduces a skip connection between the final model prediction and the past predicted frame. In the Seq2Seq-sampling-SPL approach [17], the authors introduce a Structural Prediction Layer at the decoder that results in a hierarchical prediction of joints, based on the structural prior of human joints. In addition, we compared against the zero-velocity baseline used in many other work for comparison and demonstrated to be a high-performance baseline that is hard to outperform [15], [16], [19]. The baseline assumes that all the future predictions are identical to the last observed pose and is difficult to outperform for short-term prediction.

Similar to single agent, we compared our multi-agent approach against two state-of-the-art methods, Joint Learning and Joint Learning + Social [24]. In case of Joint Learning + Social, a permutation invariant pooling mechanism is applied to pool social features across all agents with max-pooling providing the best results [24]. To ensure a fair comparison, we fine-tuned hyper-parameters for all the approaches.

C. Evaluation Metric

We evaluated the performance of all models using the Mean Squared Error (MSE), which is the l_2 distance between the ground-truth and predicted poses at each timestep, averaged over the number of joints and sequence length, similar to prior work [17], [18], [24], [32]. The MSE is calculated as:

$$\mathcal{L}(\mathcal{X}, \hat{\mathcal{X}}) = \frac{1}{T \times K} \sum_{t=1}^{T} \sum_{i=1}^{K} (x_t^i - \hat{x}_t^i)^2 \tag{10}$$

where, T and K are the total number of frame and joints respectively. The MSE jointly encodes global body motion and skeletal movements [24], making it an ideal metric.

D. Implementation Details

Our approach is divided into four modules: the encoder, latent variables, discriminators and decoder. The training has two phases: reconstruction and regularization, in line with the AAE framework [38]. In the reconstruction phase, the encoder-decoder is trained end-to-end, using reconstruction loss. In the regularization phase, the discriminators are trained using the cross-entropy loss. The discriminator loss is used to update the weights of the encoder. We provide details on the training of all experiments in the supplementary video.

Encoder: For single-agent experiments on UTD-MHAD and HRC data, we use one multi-stream encoder to encode past observations. The encoder comprises of three GRUs for position, velocity, and acceleration. The hidden state dimension is 200 for velocity and acceleration. For position, the hidden state dimension is the same as the input dimension.

For multi-agent experiments on the NTU RGB+D 60 and CMU Panoptic datasets, we varied the number of encoders depending on the number of agents. We use dropout regularization for all GRUs with a dropout probability of 0.1.

Latent variables: We empirically evaluated the ideal combination for the continuous and categorical latent variables, while ensuring that they are smaller than the intrinsic dimension of the data. The dimensions for continuous and categorical latent variables vary depending on the datasets and are provided in the supplementary video.

Decoder: For single-agent experiments, we use one decoder and implement weight sharing between the position-specific

TABLE I
MSE (In cm^2) Comparison of Different Single-Agent Methods on UTD-MHAD and KTH-HRC Datasets (Lower Is Better)

Approaches/Frames	UTD-MHAD						KTH-HRC (Trained and tested on HRC data)						KTH-HRC (Trained on HHC, tested on HRC data)					
	2	4	8	10	13	15	5	10	20	30	35	40	5	10	20	30	35	40
Zero-Velocity [16]	11.31	27.91	68.79	89.09	116.95	133.05	0.11	0.34	1.18	2.38	3.07	3.81	0.09	0.32	1.14	2.33	3.02	3.76
Seq2Seq [16]	8.90	19.09	39.03	47.45	57.84	63.30	0.18	0.55	1.67	3.11	3.91	4.74	0.14	0.36	1.09	2.17	2.81	3.49
Seq2Seq-SPL [17]	8.17	17.63	36.86	45.02	55.20	60.72	0.17	0.42	1.20	2.33	2.98	3.66	0.09	0.25	0.86	1.97	2.73	3.62
Our method	**6.39**	**14.33**	**31.63**	**39.12**	**48.57**	**53.74**	**0.06**	**0.20**	**0.72**	**1.61**	**2.21**	**2.91**	**0.07**	**0.24**	**0.78**	**1.52**	**1.95**	**2.28**

encoder and decoder GRU. The output of the GRU is followed by a Structured Prediction Layer (SPL) [17].

For multi-agent experiments, we varied the decoders depending on the number of agents. We simplify the decoder operations by using a linear layer as the final output for each decoder. In both experiments, we use Teacher Forcing [43] to aid the learning *only during training*, whereby we feed the actual output at the last timestep to prevent prediction errors from severely propagating into the future.

Discriminator: We use feedforward neural networks with 2 linear layers each, followed by sigmoid activation for both discriminators. The hidden size of both layers is 200 and 100 for single and multi-agent experiments respectively.

Training environment: We used Pytorch v1.5.1 running on Nvidia Titan v100 and Cuda 10.1 for all our experiments. The encoder-decoder architecture is trained end-to-end using the Adam optimizer [44]. We used an initial learning rate of 1e-3 for experiments on UTD-MHAD, KTH-HRC & CMU Panoptic datasets and 5e-4 for experiments on the NTU RGB+D 60 dataset. For all experiments, we used weight decay on plateau with a decay factor of 0.1 and early stopping on the validation set. For the discriminators, we used Adam optimizer with learning rates of 2e-6 onUTD-MHAD, KTH-HRC and CMU-Panoptic and 2e-7 on NTU RGB+D 60.

VI. Results and Discussion

A. Single Agent Motion Prediction

Results: We present the results of all models on single-agent motion prediction on the UTD-MHAD in Table I. We report the performance of all approaches at distinct frame intervals to circumvent the problem of frame drops during data collection and subsequent evaluation. Our frame intervals aim to evaluate all models on short (2 & 4), mid (8 & 10), and long-term motion prediction (13 & 15). The results in Table I suggest that our approach outperforms all other methods and the zero-velocity baseline for short, mid, and long-term prediction. Our proposed model performs particularly well for long-term prediction with the performance of all models deteriorating as the prediction horizon increases.

Discussion: Our proposed approach outperformed state-of-the-art models on all evaluated benchmarks, suggesting improved representation learning and sequence modeling. The results from Table I suggest that all models outperform the zero-velocity baseline [16]. For long-term motion prediction (13 & 15 frames), our method outperforms other approaches, firstly demonstrating the robust learning capability of the multistream encoder. Furthermore, the latent variables learn a distribution over the observed trajectory, which is used to predict future frames. As such, they learn long-term representation over a horizon. As the decoder conditions its output on the last predicted frame *and* the latent variables, it achieves performance gains over the long-term. Fig. 1 underscores the fact that our approach generates more feasible motion compared to other methods by accurately modeling joint position and orientation.

B. Multi-Agent Motion Prediction

Results: We present the results of all models on multi-agent motion prediction on the NTU RGB+D 60 and CMU Panoptic datasets in Table II. Similar to the single-agent setup, we measured all models' quantitative performance at the same distinct frame intervals (2, 4, 8, 10, 13 & 15). The results in Table II suggest that our approach outperforms all models over all evaluated horizons, with particularly improved performance over longer horizons.

Discussion: Our proposed approach outperformed other methods over all the evaluated horizons. This suggests that our approach learns a more robust representation for each agent, while also capturing relevant inter-agent dynamics among all the agents. The multi-stream encoder provides a salient representation for each agent, which is then used by the self-attention mechanism to adaptively weigh relevant agent-specific features for modeling the interaction dynamics among all the agents. In addition, the decoder module learns the attention weights between the immediate agent-specific past output and the latent variables representing the observed multi-agent interaction. This further contributes to the performance gain, especially over longer horizons, as the decoder conditions over a subset of past frames and interaction among all the agents.

C. Human-Robot Collaboration Experiments

Results: We present the results of all models on the human-robot collaboration experiments in Table I. We first trained and evaluated all models on HRC data. Next, we trained all models on HHC data and evaluated them on HRC data. Here, we measured the MSE over larger frame intervals due to the tasks' duration being longer (approx. 11 seconds). We evaluated all models on short (5 & 10), mid (20 & 30) and long-term horizons (35 & 40).

Discussion: The results in Table I (KTH-HRC (Trained and tested on HRC data)) suggest that our proposed method outperformed all other approaches over all the horizons. Similar to the single and multi-agent conditions, our approach's performance gains increase over longer horizons.

When training on HHC data and testing on HRC data, the results in Table I suggest a similar pattern, with our proposed approach outperforming other methods. We also observed that the models generalize better when training on HHC data and testing on HRC data. We attribute this to there being greater and more diverse training samples, which allowed the models to learn a more robust representation.

The above results demonstrate our model's ability to best predict human motion, even in the presence of a collaborative robot. Having superior short-term performance would allow the robot to prevent collisions and be more responsive, thus enhancing collaboration safety. On the other hand, having superior

TABLE II
MSE (IN CM^2) COMPARISON OF DIFFERENT MULTI-AGENT METHODS ON NTU RGB+D 60 AND CMU PANOPTIC DATASETS (LOWER IS BETTER)

Approaches/Frames	NTU RGB+D 60						CMU Panoptic					
	2	4	8	10	13	15	2	4	8	10	13	15
Joint Learning [24]	9.68	15.84	29.88	37.52	49.55	57.93	1.334	2.29	4.15	5.09	6.55	7.56
Joint Learning + Social [24]	9.71	15.97	30.36	38.25	50.70	59.38	1.396	2.39	4.35	5.35	6.87	7.90
Our method	**9.66**	**15.66**	**29.05**	**36.16**	**47.20**	**54.84**	**1.327**	**2.22**	**3.94**	**4.79**	**6.07**	**6.94**

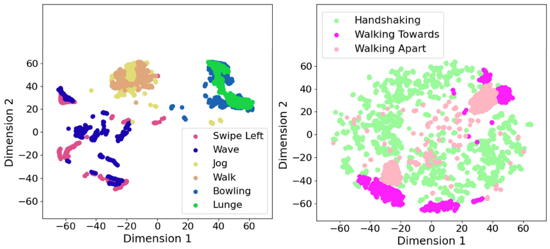

Fig. 5. Continuous latent space visualization using t-SNE plots on UTD-MHAD (Left) and NTU RGB+D 60 (Right) datasets.

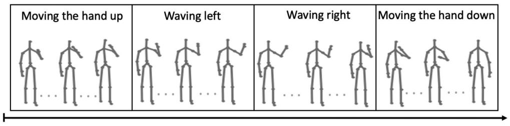

Fig. 6. Action primitives for *wave* on UTD-MHAD.

long-term performance would allow the robot to plan its actions more efficiently.

D. Latent Space Interpretation

Results: We visualized the learned latent space of our framework to improve our understanding of the generative process. For this purpose, we analyzed the learned latent manifold for single and multi-agent settings on the UTD-MHAD and NTU RGB+D 60 respectively. For each temporal window of observations, our proposed framework maps the high-dimensional data into a low-dimensional manifold, represented by the continuous and categorical latent variables.

To visualize the continuous latent variables, we project them to a 2-D plane using t-SNE [45] as shown in Fig. 5. We then segment the 2-D plane by action class labels, which were not provided during training. For analyzing the categorical latent variables, we look at their distribution over a trajectory for each action. Fig. 6 presents the predicted frames for the action *wave* on UTD-MHAD, along with the distribution of the categorical latent variable over the duration of the action.

Discussion: Our results suggest that the continuous latent variables learn spatial embedding for each temporal window. Fig. 5 shows that activities that share similar spatial semantics, such as *walking and jogging* on UTD-MHAD (Fig. 5-Left) and *walking towards and walking away* on NTU RGB+D 60 (Fig. 5-Right) have overlapping clusters. Similarly, other sets of activities such as *bowling and lunging*, and *wave and swipe left* on UTD-MHAD also have separate overlapping clusters. Additionally, as seen in Fig. 5-Left for UTD-MHAD, our framework learns to separate activities that have different spatial semantics: the clusters for *bowling and lunging*, *walking and jogging*, and *wave and swipe left*, do not overlap. Similar segmentation is observed for multi-agent activities such as *handshaking* and *walking towards/apart* on NTU RGB+D 60 (Fig. 5-Right).

TABLE III
ABLATION STUDY OF OUR METHOD ON UTD-MHAD. HERE, SPL: USING STRUCTURED PREDICTION LAYER, TF: TEACHER FORCING

Approaches/Frames	2	4	8	10	13	15
No-SPL + No-TF	8.31	17.32	35.29	43.09	52.81	57.90
No-SPL + TF	6.52	14.43	32.79	41.02	51.89	57.84
SPL + No-TF	7.61	16.14	34.29	42.27	51.97	57.01
SPL + TF	**6.39**	**14.33**	**31.63**	**39.12**	**48.57**	**53.74**

In case of categorical latent variable, our results on UTD-MHAD indicate that it takes on different values over time, which coincides with different action primitives. As can be seen in Fig. 6, the action class *wave* is segmented into four action primitives: *moving the hand up*, *waving left*, *waving right* and finally *moving the hand down*.

The above results demonstrate how our framework interprets each temporal window, modeling the continuous and categorical aspects of human motion. The learned representation of our framework can be used for various facets of robot perception, from activity segmentation and recognition to learning from demonstration. Crucially, it can be viewed as a step towards closer human-robot collaboration, by providing an explainable robot perception.

E. Ablation Study of Learning Modules

Results: We conducted an ablation study on the UTD-MHAD to evaluate the importance of various learning modules in our approach. Table III shows the impact of specific learning practices, given the same backbone framework of the multi-stream encoder and decoder.

Discussion: For a baseline, we have no SPL at the decoder, replacing it with a linear layer, while also not using Teacher Forcing (TF) during training. This architecture provided the worst performance in terms of MSE loss. Adding TF with a probability of 0.5 resulted in a large improvement in the short-term prediction, with marginal gains over the long-term. This highlights the importance of TF especially for short-term prediction while also suggesting that the benefit decreases with an increase in time. We next assess the impact of having the SPL, firstly with no TF. Consistent with previous results, the short-term performance of the model is worse when compared with No-SPL + TF; however it is better across all evaluated horizon when compared against No-SPL + No-TF. This suggests the benefit of hierarchically predicting each joint when using SPL as the final layer instead of using a linear layer that assumes all joints are independent. Our best performing model is the SPL + TF, which combines the benefit of using structured prediction as well as having the short-term improvement of TF.

VII. CONCLUSION

In this work, we introduced a novel sequence-learning approach for human motion prediction that outperformed state-of-the-art methods on single and multi-agent settings. Our framework for multi-agent motion prediction introduces an attention-based mechanism that can better represent the inter-agent dynamics of human motion. Furthermore, our framework conditions its output on a subset of past frames instead of just the last frame, thus attaining performance gains over the long-term. Finally, our approach can provide an intuitive explanation of the learned latent space. Our results suggest that the latent space can model human motion into spatial and temporal segments. The overall performance of our approach along with its interpretable latent space suggests that our approach can effectively capture the underlying dynamics of human motion. This opens the possibility of its adoption for robot perception in human-robot collaboration.

REFERENCES

[1] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, 2020.

[2] G. Hoffman, "Evaluating fluency in human-robot collaboration," *IEEE Trans. Human-Mach. Syst.*, vol. 49, no. 3, pp. 209–218, June 2019.

[3] T. Iqbal, L D. Riek, A. Goswami, and P. Vadakkepat, "Human robot teaming: Approaches from joint action and dynamical systems," *Humanoid Robot.: Ref.*, to be published, doi: 10.1007/978-94-007-7194-9_137-1.

[4] J. Bütepage, A. Ghadirzadeh, Ö. Ö. Karadag, M. Björkman, and D. Kragic, "Imitating by generating: Deep generative models for imitation of interactive tasks," *Front. Robot. AI.*, vol. 7, p. 47, 2020, doi: 10.3389/frobt.2020.00047.

[5] N. Sebanz, H. Bekkering, and G. Knoblich, "Joint action: Bodies and minds moving together," *Trends Cogn. Sci.*, vol. 10, no. 2, pp. 70–76, 2006.

[6] B. H. Repp and Y.-H. Su, "Sensorimotor synchronization: A review of recent research, 2006-2012," *Psychon. Bull. Rev.*, vol. 20, no. 3, pp. 403–452, 2013.

[7] T. Iqbal, S. Li, C. Fourie, B. Hayes, and J. A. Shah, "Fast online segmentation of activities from partial trajectories," in *Proc. Int. Conf. Robot. Automat.*, 2019, pp. 5019–5025.

[8] T. Iqbal and L. D. Riek, "A Method for Automatic Detection of Psychomotor Entrainment," *IEEE Trans. Affect. Comput.*, vol. 7, no. 1, pp. 3–16, Jan.-Mar. 2016.

[9] T. Iqbal, S. Rack, and L. D. Riek, "Movement Coordination in human-robot teams: A dynamical systems approach," *IEEE Trans. Robot.*, vol. 32, no. 4, pp. 909–919, Aug. 2016.

[10] T. Iqbal and L. D. Riek, "Coordination dynamics in multi-human multi-robot teams," IEEE Robot. Automat. Lett., vol. 2, no. 3, pp. 1712–1717, Jul. 2017.

[11] M. M. Islam and T. Iqbal, "Hamlet: A Hierarchical multimodal attention-based human activity recognition algorithm," in *IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 10285–10292 doi: 10.1109/IROS45743.2020.9340987.

[12] G. Hoffman and C. Breazeal, "Cost-Based Anticipatory Action Selection for Human-Robot Fluency," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 952–961, Oct. 2007.

[13] M. M. Islam and T. Iqbal, "Multi-Gat: A graphical attention-based hierarchical multimodal representation learning approach for human activity recognition," in *IEEE RA-L*, pp. 1–1, 2021, doi: 10.1109/LRA.2021.3059624.

[14] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5308–5317.

[15] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4346–4354.

[16] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2891–2900.

[17] E. Aksan, M. Kaufmann, and O. Hilliges, "Structured prediction helps 3d human motion modelling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7144–7153.

[18] J. Bütepage, H. Kjellström, and D. Kragic, "Anticipating many futures: Online human motion prediction and generation for human-robot interaction," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4563–4570.

[19] S. Toyer, A. Cherian, T. Han, and S. Gould, "Human pose forecasting via deep markov models," in *Proc. Int. Conf. Digit. Image Comput.: Techn. Appl.*, 2017, pp. 1–8.

[20] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 961–971.

[21] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2255–2264.

[22] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2821–2830.

[23] S. H. Park *et al.*, "Diverse and admissible trajectory forecasting through multimodal context understanding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 282–298.

[24] V. Adeli, E. Adeli, I. Reid, J. C. Niebles, and S. H. Rezatofighi, "Socially and contextually aware human motion and pose forecasting," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 6033–6040, Jul. 2020.

[25] D. Yu, H. Wang, P. Chen, and Z. Wei, "Mixed pooling for convolutional neural networks," in *Proc. Int. Conf. Rough Sets Knowl. Technol.* Springer, 2014, pp. 364–375.

[26] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 3859–3869.

[27] E. Barsoum, J. Kender, and Z. Liu, "Hp-gan: Probabilistic 3d human motion prediction via gan," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1418–1427.

[28] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5998–6008.

[29] C. Chen, R. Jafari, and N. Kehtarnavaz, "Utd-mhad: A. multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 168–172.

[30] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb d: A. large scale dataset for 3d human activity analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1010–1019.

[31] H. Joo *et al.*, "Panoptic studio: A. massively multiview system for social motion capture," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3334–3342.

[32] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6158–6166.

[33] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 336–345.

[34] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 3399–3406.

[35] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks," *Auton. Robots*, vol. 41, no. 3, pp. 593–612, 2017.

[36] V. V. Unhelkar *et al.*, "Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time," *EEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2394–2401, Mar. 2018.

[37] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Int. Conf. Learn. Representations, (ICLR)*, 2014.

[38] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *Int. Conf. Learn. Representations*, 2016.

[39] E. Dupont, "Learning disentangled joint continuous and discrete representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 708–718.

[40] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2672–2680.

[41] D. P. Losey, K. Srinivasan, A. Mandlekar, A. Garg, and D. Sadigh, "Controlling assistive robots with learned latent actions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 378–384.

[42] M. Suguitan, R. Gomez, and G. Hoffman, "Moveae: Modifying affective robot movements using classifying variational autoencoders," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interact.*, 2020, pp. 481–489.

[43] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Int. Conf. Learn Representations*, 2015.

[45] L. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.