31st International Conference on Artificial Neural Networks (ICANN), Bristol, UK, September 2022.

Intention-Aware Frequency Domain Transformer Networks for Video Prediction

Hafez Farazi and Sven Behnke

University of Bonn, Computer Science Institute VI, Autonomous Intelligent Systems
Friedrich-Hirzebruch-Allee 5, 53115 Bonn, Germany
{farazi, behnke}@ais.uni-bonn.de

Abstract. The ultimate goal of video prediction is not to predict pixel-perfect future images. Instead, it is desired to extract a valuable internal representation to solve downstream tasks. One of the essential downstream tasks is to understand the semantic composition of the scene and later use it for decision making. For example, an observer robot can anticipate human activities and collaborate in a shared workspace. However, one of the biggest challenges in human-robot collaboration remains understanding human intentions and movements. This paper focuses on predicting future frame pose activities given a pre-trained off-the-shelf pose estimation model (i.e., shelf-supervised). We propose a lightweight and interpretable model based on the Frequency Domain Transformer Networks to solve semantic prediction, given that we have multiple plausible futures. We show that the proposed model outperforms other well-known video prediction models on the pose prediction task extracted from the Human3.6M dataset and a synthetically created dataset with multiple plausible futures.

1 Introduction

Video prediction is about predicting future unseen image sequences based on some initial observed image sequence. A video prediction model that is useful in real-world scenarios should not only take into account the dynamics and content of the observed scene but also have a notion of multiple plausible futures to account for the inherent uncertainty of the dynamics of the world. For example, in human-robot collaboration scenarios, given some observed frames of a human working in a shared human-robot workspace, the robot should not only have a notion of what the most likely next movement of the human is but also what the possible future movements and intentions are. Based on this notion of multiple plausible futures, the robot can plan its following actions. Given a real-world video sequence, a classical pixel-level prediction of the video leads to a blurry prediction that represents the average of all possible futures. We argue that such a representation is only valuable for deterministic or semi-deterministic situations where the past completely or mostly determines the future. For instance, several possible futures exist when a person performs a movement in front of the camera. Suppose the person is walking in the initially observed frames. They may continue the walking sequence, decrease or increase walking speed, stop

walking, start walking backward, and many other possibilities. It is not feasible to formulate all these possible futures in advance. Instead, they must be derived from the data, preferably in a self-supervised manner. It is also essential for the future latent space to be interpretable and easy to sample from because this directly impacts how well the robot can plan its future actions.

The task we are mainly concerned with in this paper is the prediction of future semantic frames, which is a variant of video prediction. Here we are not interested in predicting the signal level but in predicting future semantics based on some observed semantics. In general, semantic prediction refers to predicting the output of another network. Therefore, the prediction model must not only learn to make predictions but also cope with imperfect seed input frames. Semantic prediction is more valuable than video prediction at the signal level because we predict the scene's essence, not just some insignificant pixel-level details. For example, in one of our experimented datasets, we predict the motion of a human subject and discard irrelevant information such as the person's hairstyle and color.

In our particular use case, the ultimate goal is to predict the human pose in a shared human-robot workspace using multiple smart edge sensors with different viewing angles. Later, we collect and fuse these short-term predictions for further processing and decision making with a more computationally powerful centralized backend server. The backend will merge the short-term local predictions into an allocentric semantic map. To enable short-term predictions, we train our model with human poses from the Human3.6M dataset [1]. The human poses were extracted using a pre-trained off-the-shelf state-of-the-art model developed by Bultmann et al. [2]. Note that the short-term prediction has to be done in real-time on the edge sensor, so models with a massive number of parameters are not suitable for this task.

Most other human skeleton prediction works are unsuitable for the described use case. Human skeleton prediction models are typically formulated as time series of 3D points corresponding to human joint positions and developed with the intention of predicting long sequences into the future [3]. Much of the recent work on this topic uses a graph neural network approach [4, 5]. These models usually cannot deal with occlusions, missing joints, and the ambiguity of human joint positions. Furthermore, they assume a perfect skeleton extractor, which is not realistic considering that the human joint extractor is also a deep learning model that looks at natural images and does not have access to the ground truth. Although such models can be used in our backend server, which has access to a fused and near-perfect 3D skeleton, they are not suitable for short-term local predictions in the sensor space, which is the purpose of this work.

Our work is an extension of our lightweight and fully interpretable FDTN-based models [6, 7] and follows our recent findings [8], which suggest that when semantic prediction is the goal, first extracting semantics followed by video prediction yields a better result compared to first performing video prediction followed by semantic extraction. We address the multiple plausible futures problem by proposing a lightweight intention model to extract very low-dimensional

stochastic latent variables, pass them to the FDTN-based prediction model, and train them jointly in an end-to-end fashion. Note that the intention model proposed in this paper is not limited to semantic predictions and can also be used for video predictions. The code and dataset of this paper will be publicly available on GitHub [1]. The main contributions of this paper are as follows:

- We extend the existing frequency-domain-based transformer models to account for multiple plausible future scenarios.
- We propose a lightweight model to extract different variants of the predictions in a very low-dimensional latent space.
- We show that the output space of our intention model is interpretable and meaningful while capturing a wide range of plausible predictions.

2 Related Work

While there are many different approaches to video prediction, the most effective ones use deep learning to create abstract representations of scene content and observed transformations. A successful example is Video Ladder Network (VLN) [9], an extension of Ladder Networks that uses a recurrent lateral link at each level and models transformations at that level of abstraction, with the lowest level representing video frames. Conversely, PredRNN++ [10] consists of a stack of LSTM modules, with the output of each module fed into the subsequent module, forming a frame prediction at the top. PredNet [11], which aims to improve neural plausibility, implements a hierarchical architecture that learns a generative model of the input per layer. Only deviations from the expected input are propagated upward, actualizing the concepts of predictive coding. In an extension of this idea, HPNet [12] also resorts to associative coding and adds a direct upstream of spatio-temporal feature encodings extracted by 3D convolution. Here, the feedback path is routed to an LSTM at each level. In addition to generating plausible future images, the above two ideas highlight the exciting potential of video prediction tasks in studying models of cortical processing. In contrast, other approaches mostly ignore image content and focus on the dynamics of the scene. For example, PGP [13, 14] integrates a gated autoencoder and the transformation model of RAE [15] to learn encodings of global linear image transformations between successive frames.

Most of the existing video prediction models are suitable for deterministic datasets, do not fully capture the distribution of outcomes, and provide blurry predictions in stochastic datasets. The blurry prediction is the result of not explicitly modeling multiple plausible futures, so the model is forced to produce the aggregate of multiple predictions to reduce the loss. Recently, loss functions that specify a distribution of the outcome have been explored. One such approach is the adversarial loss [16]. Still, the difficulty of training, the overhead of using a discriminator, and the mode collapse make GAN-based approaches not ideal for real-time video prediction. Variational inference is another solution to the problem. While there are few previous works dealing with multiple plausible futures

[1] `https://github.com/AIS-Bonn/Intention-Aware-Video-Prediction`

in the form of variational inference frameworks [17,18], they are often difficult to train and require a complex freezing and training scheme [18]. Moreover, these models typically require a high-dimensional stochastic latent space, complicating their interpretation because it is impossible to gain insight by exploring the fully formed latent space. Finally, these models often use hundreds of random samples in hopes of finding the best matching future prediction when reporting their test performance.

We argue that the desired model should use a very low-dimensional latent space with a known range while generating a diverse future outcome. Furthermore, if the latent variables have low dimensionality with a known range, we can iterate through the latent space and gain insight into the model's predictive ability. Finally, it is also desirable to group the latent variable across multiple frames to force it to form a concept of the variability of the possible motions, rather than encoding meaningless jitter-like variations on each prediction frame. In the following sections, we propose our model that meets these criteria.

3 Models

In this section, we introduce the components used in our experiments.

Local Frequency Domain Transformer Networks (LFDTN): The core functionality of Local Frequency Domain Transformer Networks is the ability to describe changes in an observed image like signal as a collection of local linear transformations, transport inferred shifts into the future, and consequently apply them to make a prediction for the content of the next frame. The first part of these three distinct tasks is performed by a process that can be described as *Local Fourier Transform* (LFT), a Fourier-based transform similar to STFT for a 2D signal. For a given image x_t, overlapping tiles are extracted and windowed with a function w to produce $x_{t,u,v}$, a collection of tapering windows on x_t around the image coordinates $\{u, v\}$. For each, the FFT $\mathcal{X}_{t,u,v}$ is computed. For $\mathcal{X}_{t-1,u,v}$ and $\mathcal{X}_{t,u,v}$ (the LFTs of two consecutive images x_{t-1} and x_t), the *local phase difference* is then defined element-wise as:

$$\mathcal{PD}_{t-1,u,v} \coloneqq \frac{\mathcal{X}_{t,u,v}\overline{\mathcal{X}_{t-1,u,v}}}{|\mathcal{X}_{t,u,v}\overline{\mathcal{X}_{t-1,u,v}}|}. \tag{1}$$

The local phase differences encode the image shift observed around $\{u, v\}$ and serve as a content-independent description of the local image transformation. Since local adversities sometimes perturb the phase differences, in addition to the fact that the shifts are generally not spatiotemporally constant, a lightweight learnable convolutional network $\mathcal{MM}$ filters and transports them one time-step ahead. We call this the "transform model" and apply it as:

$$\widehat{\mathcal{PD}}_{t,u,v} = \mathcal{MM}(\mathcal{PD}_{t-1,u,v}). \tag{2}$$

It should be noted that $\mathcal{MM}$ was designed with an intentional bottleneck that forces the representation of $\widehat{\mathcal{PD}}_{t,u,v}$ as a vector field in the output layer that can be easily accessed and that can well explain the final prediction results.

Next, a prediction of the local views on x_{t+1} is formed via the *local phase addition* given by :

$$\widehat{\mathcal{X}}_{t+1,u,v} = \mathcal{X}_{t,u,v} \cdot \widehat{\mathcal{PD}}_{t,u,v} \tag{3}$$

and subsequently to obtain their inverse Fourier transforms $\hat{x}_{t+1,u,v}$. In addition, the effects of local displacements on the tapering windows are considered by repeating this step for the Fourier transform of the window function $\mathcal{W}$:

$$\hat{w}_{t,u,v} \coloneqq \text{iFFT}(\text{phase\_add}(\mathcal{W}, \widehat{\mathcal{PD}}_{t,u,v})). \tag{4}$$

Using both, the next video frame x_{t+1} is reconstructed by inverse local Fourier Transform in the presence of shifted windows. The sequence of analysis, then transport and prediction, and finally synthesis described above can also be applied on a channel-by-channel basis. This means that any spatial signal, e.g., a segmented video or human keypoints activity maps, is also a valid input. For more details on LFDTN, we encourage the reader to read the original LFDTN paper [6].

Global Frequency Domain Transformer Networks (GFDTN): Here we present a special case of LFDTN, which for clarity, we call Global Frequency Domain Transformer Networks (GFDTN). This model is similar to LFDTN and analyzes the signal globally. We set the size of the analysis window to the full input resolution and replace the window function with the identity. Because the LFDTN uses positional encoding channels, it can infer the location of each local transform and use it for prediction. Since in GFDTN, we only have a global window, the location-dependent features are not needed. To compensate for this, we replace the positional encodings with each input channel's Center of Mass.

Evidently GFDTN is limited and can only model one global transformation per specified channel. Consequently, although using GFDTN cannot make predictions at the signal level in natural videos, it can predict simple signals, such as blob-like semantics separated in different channels, into the future. Furthermore, due to the rigid assumption of a single global motion, this model converges faster than LFDTN. When the difference between LFDTN and GFDTN is not the focus of discussion, we refer to both models as FDTN in this paper.

Intention Aware Network: The core idea behind this model is to encode different variations in the dataset and produce a latent variable, which we call the $\boldsymbol{z}$ vector. During training, this model has access to the future frames to extract the essence of the future changes. We then feed the $\boldsymbol{z}$ vector into the FDTN model by concatenating $\boldsymbol{z}$ as an additional channel to the "transform model". The $\boldsymbol{z}$ vector is a very low-dimensional representation of these future variations that helps the prediction model decouple stochasticity from prediction. Unlike the SVG-LP model [17], which computes the posterior distribution for each time step, the $\boldsymbol{z}$ latent variable is time-invariant, i.e., it encodes the variations once for all future frames. Note that during testing, instead of computing the $\boldsymbol{z}$ vector using the intention model, we can iterate through different possible $\boldsymbol{z}$ values to generate diverse prediction frames. The experiments section shows that we can generate different plausible predictions by changing the $\boldsymbol{z}$ variable. The $\boldsymbol{z}$ latent space can be continuous or discrete. The discrete version is motivated by the

recent success of discrete generative models like video VQ-VAE [19]. Note that one can utilize both discrete and continuous latent variables simultaneously.

The intention model is a copy of the "transform model" with some modifications. Some functions, such as the extraction of local phase differences, are computed once between each successive frame and then used in both the prediction and intention models to avoid redundant computations. The last layer is the difference between the "transform model" in FDTN and the intention model. The intention model's last layer is a fully-connected layer followed by non-linearity that outputs $\boldsymbol{z}$. In the continuous space, the nonlinearity is a tanh activation, and in the discrete case, it is a softmax layer.

In contrast, in the FDTN, the last layer is a convolutional layer that generates two-dimensional vectors that are then used in the phase addition process. Due to the similarity between the "transform model" in FDTN and the intention model, and to save learnable parameters and speed up the training process, the FDTN model shares weights with the intention model in the human joint prediction experiments. We also experimented with replacing the affine layer in the intention model with a global average pooling layer, and the results were slightly worse, so we decided to use the affine layer.

As it can be seen in the Fig. 1, without this model, the FDTN model, when faced with a stochastic dataset, will produce a blurry outcome which is a superposition of all different variations. Finally, note that the idea of an intention-aware network is not limited to FDTN based models and can also be applied to other deterministic video prediction models to enable them to model multiple plausible futures.

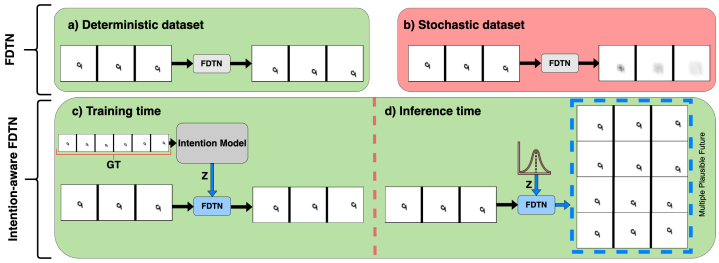


Fig. 1: An overview of the difference between the original FDTN model and the proposed intention-aware FDTN. a) The FDTN model is generating sharp future frames when trained on the deterministic "Moving MNIST" dataset. b) The FDTN model is producing a blurry result on the stochastic version of "Moving MNIST" dataset. c) The proposed intention-aware model during training. d) The proposed model while testing. Note that we can generate multiple plausible futures by sampling multiple times from the $\boldsymbol{z}$ vector in test time.

4 Experimental Results

Datasets and Training: We use a variant of the synthetic Moving MNIST dataset to evaluate our proposed architecture and show the effect of stochasticity in video prediction. We call it "Stochastic Moving MNIST". It contains six

frames with one MNIST image moving within a 64×64 frame. After three seed frames, the digit can maintain the initial velocity direction or reverse the velocity in X and/or Y directions. The choice between these four discrete random variants is made uniformly. This dataset is ideal for studying multiple plausible future scenarios with finite options.

We also created another dataset for the 2D human pose prediction task. We extracted this dataset from the Human3.6M dataset [1] using an off-the-shelf pretrained model by Bultmann et al. [2]. The "Human3.6M Joints" dataset contains a sequence of ten frames with the shape 64×64. Each frame has 14 channels representing different human joints. The time difference between each frame is about $100ms$, and we predict the last five frames given the first five seed frames. This dataset has a very diverse range of actions captured from many different human subjects with multiple camera viewing angles.

Our models were trained end-to-end using backpropagation through time. In addition, we used AdamW optimizer and MSE prediction loss. For training the discrete intention model, we scheduled the softmax temperature by starting with a high temperature and gradually decreasing it to a very low temperature simulating one-hot-encoded argmax.

Table 1: Results for "Stochastic Moving MNIST" and "Human3.6M Joints" datasets.

	"Stochastic Moving MNIST"				"Human3.6M Joints"			
Model	L1	MSE	DSSIM	Params	L1	MSE	DSSIM	Params
Conv-PGP [14]	0.01318	0.00507	0.06239	313K	0.00074	0.00015	0.00380	**640K**
HPNET [12]	0.01330	0.00489	0.06140	1.5M	0.00093	0.00020	0.00585	12.3M
SVG-LP-1D-$Best_{100s}$ [17]	0.00390	0.00102	0.01016	12.6M	0.00105	0.00023	0.00531	12.6M
SVG-LP-10D-$Best_{100s}$ [17]	0.00451	0.00136	0.01233	12.6M	0.00148	0.00037	0.00903	12.6M
SVG-LP-10D-$Best_{1000s}$ [17]	0.00321	0.00062	0.00717	12.6M	0.00144	0.00035	0.00881	12.6M
Our-GFDTN-Det	0.01347	0.00503	0.06497	**45K**	0.00064	0.00012	0.00296	**390K**
Our-LFDTN-Det	-	-	-	-	0.00071	0.00015	0.00344	**390K**
Our-GFDTN-C1D-$Best_{11s}$	0.00339	0.00133	0.00695	**78K**	**0.00057**	**0.00011**	**0.00249**	**390K**
Our-GFDTN-C1D-$Best_{21s}$	0.00175	0.00038	0.00239	**78K**	**0.00055**	**0.00010**	**0.00241**	**390K**
Our-LFDTN-C1D-$Best_{11s}$	-	-	-	-	0.00058	0.00012	0.00258	**390K**
Our-LFDTN-C1D-$Best_{21s}$	-	-	-	-	**0.00057**	**0.00011**	0.00251	**390K**
Our-GFDTN-D4	**0.00067**	**0.00004**	**0.00033**	82K	-	-	-	-
Our-GFDTN-D6	**0.00059**	**0.00003**	**0.00025**	85K	0.00061	0.00012	0.00278	**390K**
VLN-ResNet [20]	0.01220	0.00467	0.05536	1.3M	0.00078	0.00014	0.00368	1.3M
VLN-LDC [21]	0.01241	0.00470	0.05565	1.3M	0.00078	0.00014	0.00366	1.3M
PredRNN [22]	0.01142	0.00434	0.05173	1.8M	0.00070	0.00015	0.00345	3M
PredRNN++ [10]	0.01167	0.00431	0.05222	2.8M	0.00070	0.00014	0.00344	4M
Copy last frame	0.01440	0.01000	0.04105	-	0.00085	0.00028	0.00441	-
SVG-LP-1D-*Approx* [17]	0.00289	0.00043	0.00628	12.6M	0.00099	0.00021	0.00464	12.6M
SVG-LP-10D-*Approx* [17]	0.00275	0.00038	0.00594	12.6M	0.00080	0.00011	0.00303	12.6M
Our-GFDTN-C1D-*Approx*	0.00090	0.00008	0.00055	78K	0.00055	0.00010	0.00238	390K
Our-LFDTN-C1D-*Approx*	-	-	-	-	0.00056	0.00011	0.00249	390K

Evaluation: We compared our model against many well-known models, including Conv-PGP [14], VLN-ResNet [20] VLN-LDC [21], HPNet [12], PredRNN [22], PredRNN++ [10], and SVG-LP [17]. We also showed the result if we simply copy the last seed frame. Table 1 reports the outcomes on the "Stochastic Moving MNIST" and on the "Human3.6M Joints" datasets. Fig. 2 and Fig. 4 depict two sample results on each dataset for each tested baseline.

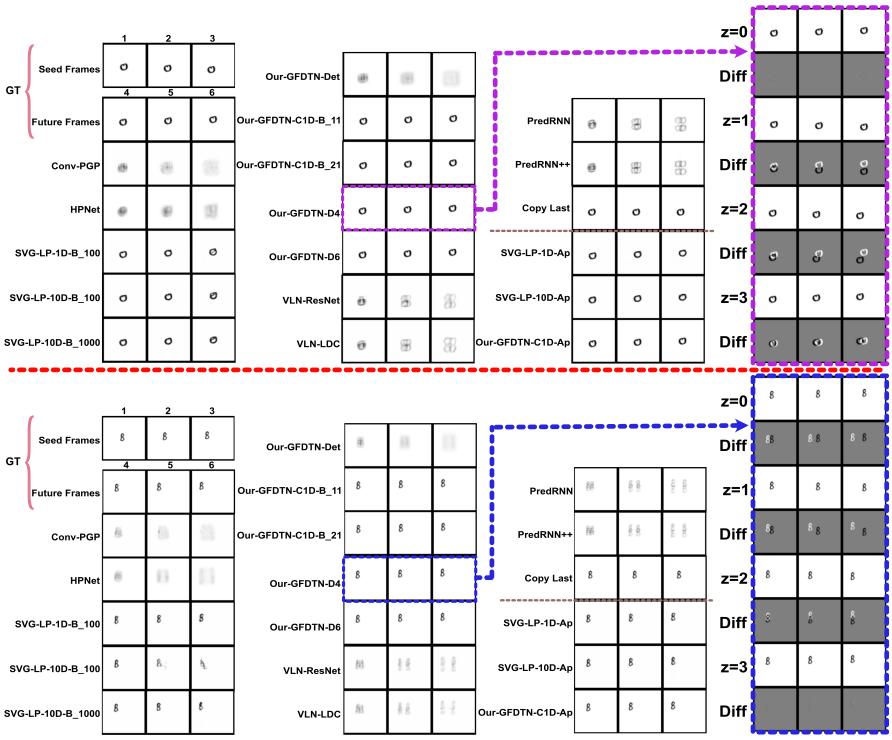
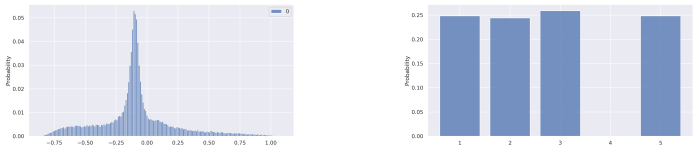


Fig. 2: Two sample results on the "Stochastic Moving MNIST" dataset were tested on different baselines. On the right, multiple plausible future predictions are generated by using different discrete choices of $\boldsymbol{z}$ on the Our-GFDTN-D4 model. The variations are plausible and reflect the stochasticity of the dataset.

In these tables for SVG-LP [17] model, **1D** and **10D** indicate the dimensionality of the used latent space. At the same time, **Approx** means that the posterior is computed by accessing future frames, and $\textbf{Best}_{\textbf{100s}}$,$\textbf{Best}_{\textbf{1000s}}$ means that we draw random samples from the latent space 100 and 1000 times, respectively, and report the best loss. For our models, **C1D** means that the $\boldsymbol{z}$ vector has dimensionality one and is continuous, while **D6** and **D4** means that the $\boldsymbol{z}$ vector is discrete with 6 and 4 discrete choices, respectively. The word **Det** represents the deterministic version of the FDTN models without using the intention model. In our models, $\textbf{Best}_{\textbf{11s}}$ and $\textbf{Best}_{\textbf{21s}}$ mean that we can obtain these results by iterating through the $\boldsymbol{z}$ vector with the range [-1,1] with a fixed interval of 0.2 and 0.1 respectively and report the best test loss. Note that all results with the word **Approx** require access to the future test images and are therefore not realistic. We reported them here to compare the results among themselves but not with other baselines and also to show the best possible results when the sample size approaches infinity. Unlike the SVG-LP model, we can easily show the different choices for the $\boldsymbol{z}$ vector by iterating through discrete choices or by choosing a fixed interval. Two example results for different $\boldsymbol{z}$ choices are shown in Fig. 2 for the synthetic dataset, which indicates that the model successfully

captures all four possible plausible futures. Furthermore, Fig. 4 depicts different $\boldsymbol{z}$ values which exhibit that the model has arranged the latent variable $\boldsymbol{z}$ in a highly interpretable and organized manner, which is a direct consequence of using a bounded low-dimensional latent variable.



(a) Our-GFDTN-C1D-*Approx* (b) Our-GFDTN-D6

Fig. 3: The histogram of the learned $\boldsymbol{z}$ vector during training. a) Is the result of the continuous $\boldsymbol{z}$ vector in the Our-GFDTN-C1D-*Approx* model, trained on "Human3.6M Joints". b) Is the result of discrete $\boldsymbol{z}$ for the Our-GFDTN-D6 model trained on "Stochastic Moving MNIST". Note that the discrete index of zero and four is not utilized by the model.

We can constrain our choices during inference time by examining the gathered histogram of the $\boldsymbol{z}$ vector during training. Two example histograms are shown in Fig. 3. It is to be observed that, in Fig. 3:b, the model with six discrete options only utilized four of the available options. This is due to a small L2 regularization used during training to encourage fewer choices. Since it is not always possible to determine the exact dimensions of the discrete variables in advance, we can choose a large enough dimension and force the model to select a minimal number of choices by increasing the regularization term.

We specified four metrics, including L1, MSE, and DSSIM, to compare our model against other baselines. Overall, experimental results indicate that our models perform well compared to other deterministic and stochastic models. We require much fewer sample iterations than SVG-LP while obtaining better results. Moreover, we need very few learnable parameters because we share the weights between the intention model and FDTN and also use lightweight FDTN models. Although we experimented with both the discrete and continuous versions of our intention model on both datasets, it is evident from the results that the discrete model works best in the synthetic dataset, which has clear and distinct plausible futures. On the other hand, the continuous model works best in the "Human3.6M Joints" dataset, where stochasticity is inherently continuous. Note that continuous variable models are easier to train compared to discrete latent variables, mainly because the gradient flows much better in the continuous version. Also, temperature scheduling is a critical part of training, and the range of the parameter and the decay rate is not a trivial hyper-parameter to tune.

SVG-LP is a model developed to address multiple plausible futures using a variational inference framework that requires a prior distribution. However, the Gaussian prior in SVG-LP, which is enforced by an additional KLD loss, is a strong assumption and leads to inferior results when the stochasticity in

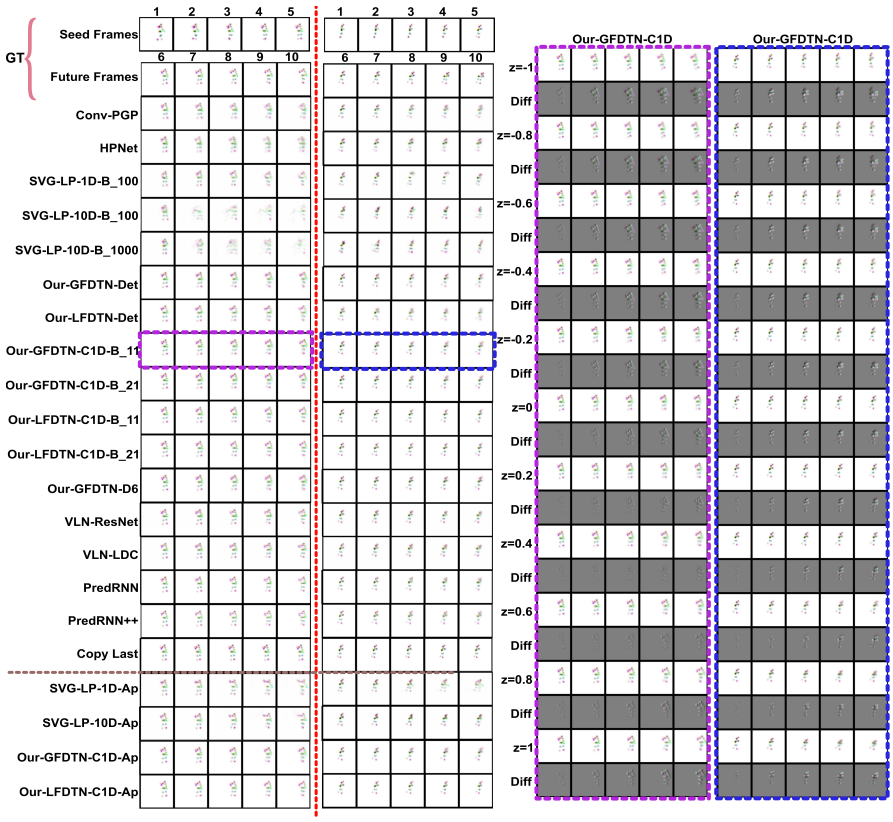


Fig. 4: Two sample results on the "Human3.6M Joints" dataset which were tested on different baselines. Right two columns show multiple plausible future predictions, given different values of $\boldsymbol{z}$ on the Our-GFDTN-C1D model. Note that all predictions are plausible, and the $\boldsymbol{z}$ dimension is organized in an interpretable manner. For example, changing the $\boldsymbol{z}$ vector can produce different walking directions in the left sample and change the amount of bending in the right.

the dataset has a different distribution. For instance, the uniformly distributed stochasticity in our synthetic dataset makes Gaussian prior a problem for the SVG-LP model. To remedy this, we utilized a lower β hyperparameter than the value originally proposed in the paper. In addition, time-variant stochasticity makes it unrealistic to iterate all possible plausible futures in SVG-LP. The required large dimensions of the latent variable combined with the time-variant stochasticity enable the SVG-LP model to essentially memorize the predictions and leak the future frames into the latent vector, leading to good training loss but inferior test performance. Although SVG-LP produces multiple plausible futures and generally outperforms most basic deterministic models after sampling multiple times and reporting the best result, the number of samples required is very high due to two main reasons. One reason is that the range of latent variables is not strictly bounded and is merely enforced with KLD loss. The second reason is that latent variables are generated per sample, which exponen-

tially increases the number of samples required. Our proposed model does not have these problems. Therefore, we can achieve a diverse and plausible prediction with very few samples. Another problem with the SVG-LP model is that many aspects of the prediction, including the motion and content, are entangled in multiple LSTM layers. Hence the latent variable that is supposed to capture the stochasticity of the motions may also encode some variations of the content shapes (see Fig. 2 for an example of this problem). On the other hand, in our FDTN models, motion and content are clearly separated, so shape variations do not contaminate the motion stochasticity.

A fixed iteration interval works very well in our experimented datasets. Nevertheless, one can use k-means clustering in training time to find K clusters and iterate over the midpoints of the clusters at inference time. Although more sophisticated sampling methods such as beam search or top-k sampling can be used depending on the computational budget, the maximum gain would not exceed **Approx** methods that have access to the approximated latent variables given the future frames. Note that if a bell curve-like shape is required in the $\boldsymbol{z}$ variables, we can add an explicit regularization loss computed on the batch of $\boldsymbol{z}$. On the other hand, if uniform distribution is desired, it can be enforced by a suitable additional loss term such as the label smoothing loss.

Our GFDTN model was successfully deployed to the Nvidia Jetson Xavier NX board, and we ran it in parallel with the human keypoint extraction model on the same board. We achieved about $8 \sim 10Hz$ for single-person semantic prediction on this computationally limited GPU.

5 Conclusion and Future Work

We proposed Intention-aware Frequency Domain Transformer Networks (IFDTN), a fully interpretable and lightweight differentiable model for the video and semantic prediction tasks. The intention network encodes the stochasticity of the dataset in a vector with very low dimensionality. By multiple sampling of the latent space, we can generate a diverse set of plausible predictions. The latent representation formed is highly organized and interpretable. Furthermore, our models require very few learnable parameters, making them highly generalizable to unforeseen data. Experiments with synthetic data and human joints extracted from real data indicate that our models can outperform other baselines with far fewer parameters. In the future, we would like to fuse our short-term predictions with the semantic extraction model to improve the overall performance of semantic extraction.

Acknowledgment: This work was funded by grant BE 2556/16-2 (Unit FOR 2535 Anticipating Human Behavior) of the German Research Foundation (DFG). The authors would like to thank the open-source community and Mark Prediger for providing baseline methods and Simon Bultmann for providing the "Human3.6M Joints" dataset.

References

1. C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.
2. S. Bultmann and S. Behnke, "Real-time multi-view 3D human pose estimation using semantic feedback to smart edge sensors," in *RSS*, 2021.
3. A. Hernandez, J. Gall, and F. Moreno-Noguer, "Human motion prediction via spatio-temporal inpainting," in *ICCV*, 2019.
4. Q. Cui, H. Sun, and F. Yang, "Learning dynamic relationships for 3D human motion prediction," in *CVPR*, 2020.
5. M. Li, S. Chen, Y. Zhao, Y. Wang, and Q. Tian, "Dynamic multiscale graph neural networks for 3D skeleton-based human motion prediction," in *CVPR*, 2020.
6. H. Farazi, J. Nogga, and S. Behnke, "Local frequency domain transformer networks for video prediction," in *IJCNN*, 2021.
7. H. Farazi and S. Behnke, "Frequency domain transformer networks for video prediction," in *ESANN*, 2019.
8. H. Farazi, J. Nogga, *et al.*, "Semantic prediction: Which one should come first, recognition or prediction?," 2021.
9. F. Cricri, X. Ni, M. Honkala, E. Aksu, and M. Gabbouj, "Video ladder networks," *CoRR*, vol. abs/1612.01756, 2016.
10. Y. Wang, Z. Gao, M. Long, J. Wang, and P. S. Yu, "PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning," in *ICML*, 2018.
11. W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *arXiv preprint arXiv:1605.08104*, 2016.
12. J. Qiu, G. Huang, and T. Lee, "A neurally-inspired hierarchical prediction network for spatiotemporal sequence learning and prediction," *arXiv preprint arXiv:1901.09002*, 2019.
13. V. Michalski, R. Memisevic, and K. Konda, "Modeling deep temporal dependencies with recurrent grammar cells," in *NeurIPS*, 2014.
14. F. D. Roos., "Modeling spatiotemporal information with convolutional gated networks," Master's thesis, Chalmers University of Technology, 2016.
15. R. Memisevic, "Learning to relate images: Mapping units, complex cells and simultaneous eigenspaces," *ArXiv*, vol. abs/1110.0107, 2011.
16. Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle GAN," in *CVPR*, 2019.
17. E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *ICML*, 2018.
18. M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," *arXiv preprint arXiv:1710.11252*, 2017.
19. W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas, "VideoGPT: Video generation using VQ-VAE and transformers," *arXiv preprint arXiv:2104.10157*, 2021.
20. F. Cricri, X. Ni, M. Honkala, E. Aksu, and M. Gabbouj, "Video ladder networks," *arXiv:1612.01756*, 2016.
21. N. Azizi, H. Farazi, and S. Behnke, "Location dependency in video prediction," in *ICANN*, 2018.
22. Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal lstms," in *NPIS*, 2017.