Imitative Non-Autoregressive Modeling for Trajectory Forecasting and Imputation

Mengshi Qi*[1,2], Jie Qin[4], Yu Wu[2,3], and Yi Yang[3]

[1]CVLab, EPFL, Switzerland [2] Baidu Research, China
[3]ReLER, University of Technology Sydney, Australia
[4]Inception Institute of Artificial Intelligence, UAE
mengshi.qi@epfl.ch; qinjiebuaa@gmail.com; yu.wu-3@student.uts.edu.au; yi.yang@uts.edu.au

Abstract

Trajectory forecasting and imputation are pivotal steps towards understanding the movement of human and objects, which are quite challenging since the future trajectories and missing values in a temporal sequence are full of uncertainties, and the spatial-temporally contextual correlation is hard to model. Yet, the relevance between sequence prediction and imputation is disregarded by existing approaches. To this end, we propose a novel imitative non-autoregressive modeling method to simultaneously handle the trajectory prediction task and the missing value imputation task. Specifically, our framework adopts an imitation learning paradigm, which contains a recurrent conditional variational autoencoder (RC-VAE) as a demonstrator, and a non-autoregressive transformation model (NART) as a learner. By jointly optimizing the two models, RC-VAE can predict the future trajectory and capture the temporal relationship in the sequence to supervise the NART learner. As a result, NART learns from the demonstrator and imputes the missing value in a non-autoregressive strategy. We conduct extensive experiments on three popular datasets, and the results show that our model achieves state-of-the-art performance across all the datasets.

1. Introduction

Owing to the increasing demands of emerging artificial intelligence applications, trajectory forecasting and imputation have surged as the intriguing topics in related fields. As illustrated in Figure 1, the sequence prediction task refers to generating the future trajectory given the previous movement, while sequence imputation aims to fill the missing value in a sequence based on a few discrete observations.

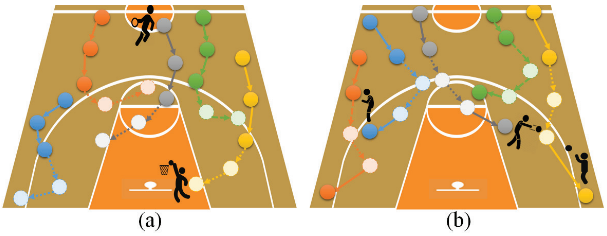


Figure 1. Trajectory forecasting and imputation. Given a 2D trajectory history of several moving agents in a basketball match, we need to (a) predict the future trajectories (*i.e.*, given a few prior continuous observations, predicting the next values), and (b) impute the missing values (*i.e.*, given some discrete observations, imputing the missing values). Circles and lines of different colors refer to the position and movement of various players, where the light-colored circles and dotted lines represent the missing values.

A large number of real-world issues can be leveraged as the tasks of trajectory forecasting and imputation, such as sports analysis [10, 23, 33], motion capture [37, 39], scene understanding [32, 34], autonomous driving [25, 49], *etc*. For example, robots or autonomous vehicles usually need to predict the future movement of persons or cars, and then promptly give feedback or take action. Meanwhile, due to trajectory crossing or occlusion, how to reason the missing value in a temporal sequence also plays a significant role in many crucial applications. Hence, adding the capability of foreseeing what will happen and complement the missing value in a sequence are imperative. Unfortunately, so far, there is still a lack of promising solutions to these problems.

Several efforts have been devoted into the tasks of trajectory forecasting and imputation [1–3, 10, 13, 24, 51], of which some methods adopt generative models (*e.g.*, variational autoencoder (VAE) [10] or generative adversarial networks (GAN) [29, 46]) to generate future movement and missing value, while others utilize intermediate variables

[*] Part of this work was done when M. Qi was visiting Baidu Research.

or representations to capture long-term coordination by hierarchical models [48, 51]. However, most of them typically focus on learning coarse disentangled latent variables, neglecting the importance of extra fine-grained conditions in the environment (*e.g.*, context and identity). Moreover, most current deep generative models are autoregressive, *i.e.*, they model the value at the current time step depending on the prior motion. Hence, they usually suffer from compounding errors in long-range temporal sequence modeling.

Furthermore, there is currently no unified framework that simultaneously handles trajectory forecasting and imputation. Although tremendous accomplishments have been achieved to address such two tasks separately, all of the existing studies fail to take the relevance between two tasks into consideration. In fact, sequence prediction can provide the reasonable temporal dependence between the previous and the future, which is also essential and significant for imputing the missing value in a temporal sequence. Thus, the aforementioned studies might be compromised due to this point. In addition, how to transfer the beneficial spatial-temporal information from the prediction model to help supervise the imputation model remains an open challenge.

To address the aforementioned issues, we propose a novel imitative non-autoregressive modeling framework for future trajectories prediction and missing value imputation. Specifically, we present a recurrent conditional variational autoencoder (RC-VAE) to capture regressive relevance in the sequence and generate future movement based on prior observations. Meanwhile, we introduce a non-autoregressive transformation model (NART) to translate an incomplete sequence into a complete sequence. Moreover, our key insight is to leverage an imitative learning strategy in the whole model, which especially inserts the imitation learning module into each part and enables RC-VAE as a demonstrator to supervise the NART learner. It is also noteworthy that our framework can transfer the spatial-temporal relevance through regressive modeling into a non-autoregressive model by virtue of such a strategy. Our main contributions are summarized as follows:

1) We propose a novel imitative non-autoregressive modeling approach to simultaneously addressing trajectory prediction and missing value imputation.

2) We introduce a new recurrent forward-backward architecture and a non-autoregressive inference strategy to learn long-term dependency and model the temporal distribution, respectively, which enable the proposed model to generate future trajectories and impute missing values.

3) We adopt the idea of imitation learning to further enhance our proposed model, through a demonstrator to supervise each decoding state of the learner.

4) Extensive experiments on three widely-adopted challenging benchmarks verify the superior performance of the proposed model over the state-of-the-art methods.

2. Related Work

Trajectory Prediction and Imputation. How to predict future motion trajectory [2, 4, 10, 17, 18, 21, 24, 27, 30, 42, 48, 50, 51] and impute missing value [6, 13, 29, 46, 47] in sequences are very important yet challenging tasks. For instance, Zheng *et al.* [51] proposed a deep hierarchical policy model and Felsen *et al.* [10] utilized conditional variational autoencoders to predict fine-grained multi-agent motions. Previous studies of missing value imputation focused on utilizing statistical methods, such as linear regression [3] and k-nearest neighbors [13]. Recent researchers utilized deep generative models to impute time series, *e.g.*, recurrent neural networks [6, 47] and generative adversarial networks [29, 46]. However, such autoregressive models have weaknesses in matching the generated data and the actual values. In this work, we propose to leverage the imitation learning into a non-autoregressive model to fulfill sequence prediction and imputation.

Imitation Learning. Recently, there have been several major advances in imitation learning [11, 19, 25, 35, 38, 43, 52], showing strong learning abilities to mimic an agent's behavior from a set of demonstration. Ziebart *et al.* [52] and Finn *et al.* [11] solved imitation learning with the inverse reinforcement learning. Ho *et al.* [19] incorporated imitation learning into adversarial training. In this work, we are the first to propose a non-autoregressive model with imitation learning for sequence prediction and imputation.

Generative Models. Nowadays, generative adversarial networks (GAN) [14] and variation autoencoder (VAE) [22] have already captured increasing attention. Especially, VAE generates new examples by regularizing the latent variable for a mass of applications, such as image generation [15], sentences generation [5], and predicting future motions in static or dynamic scenes [24, 41]. Very recently, non-autoregressive models were also introduced to related applications (*e.g.*, natural language processing [16, 26, 44] and speech synthesis [31]) because of their competitive advantages over autoregressive methods. We present a joint framework with imitation learning to address trajectory prediction and imputation, which can simultaneously take both advantages of autoregressive and non-autoregressive models.

3. Preliminaries

Problem Definition. We define the problem of multi-agent trajectory generation as follows. Given an observed set Ω of agents over the time $[t_0, t_q]$ and their corresponding trajectories $X_\Omega^{[t_0,t_q]} = X_i^{[t_0,t_q]}|_{\forall i \in \Omega}$. And the trajectory history of the i-th agent can be defined as $X_i^{[t_0,t_q]} = \{x_i^{t_0}, x_i^{t_0+1}, \cdots, x_i^{t_q}\}$, where each trajectory x_i^t is represented as the 2D coordinates at time t. The goal of our task is to generate or predict the future motion trajec-

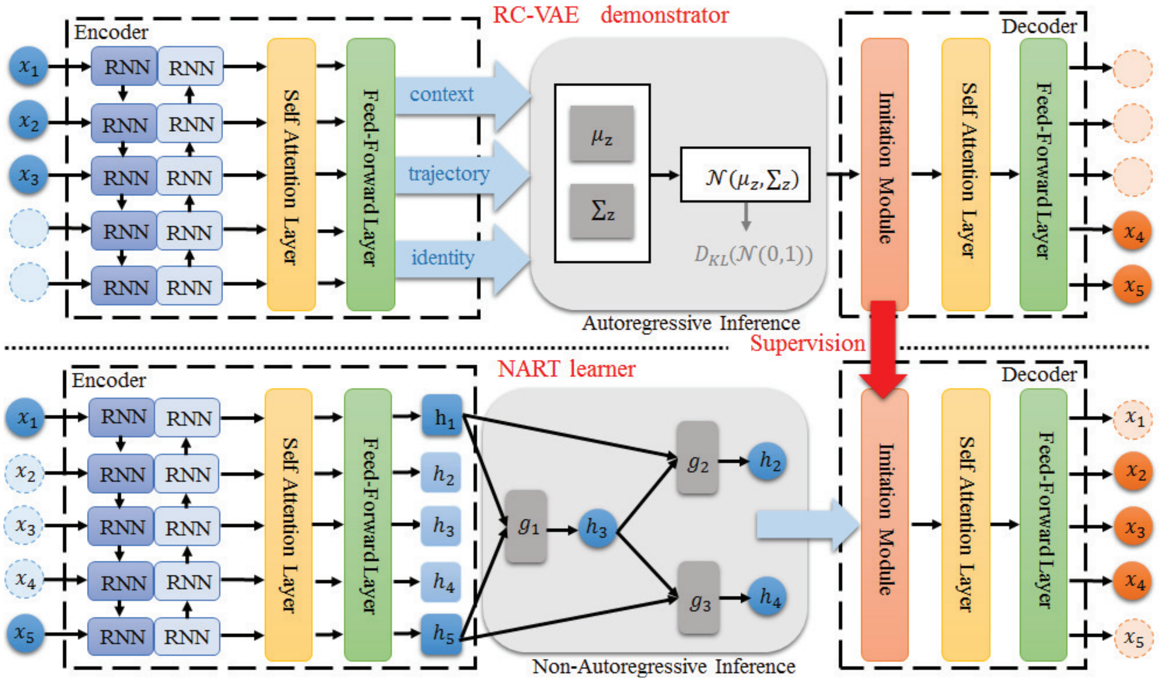


Figure 2. Overview of the proposed Imitative Non-Autoregressive Modeling framework. Our model mainly consists of two parts: the upper part is a *Recurrent Conditional Variation Autoencoder (RC-VAE) demonstrator* that is utilized to predict the future motion given an observed prior sequence, and the bottom part is a *Non-Autoregressive Transformation Modeling (NART) learner* that is leveraged to impute the missing value given a few discrete observation. Notably, we introduce the imitation modules in two models, which enable the RC-VAE demonstrator to supervise the NART learner in the decoding process.

tory of multiple agents from time t_{q+1} to time t_v, that is to learn the posterior distribution $P(Y_{\mathcal{P}}^{(t_q,t_v]}|X_{\Omega}^{[t_0,t_q]}, \mathcal{O})$, where $Y_{\mathcal{P}}^{(t_q,t_v]} = \{Y_j^{(t_q,t_v]}\}|_{\forall j \in \mathcal{P}}$ refers to the future trajectory of a subset $\mathcal{P}$ of agents, $\mathcal{P} \in \Omega$, and $\mathcal{O}$ means the other available input conditions. Following [10], we utilize the identities and the future context as the extra conditions $\mathcal{O}$, of which 'context' denotes movements of *defensive* players when predicting the future trajectories of *offensive* players, and 'identity' indicates the team that each agent belongs to and the role (*i.e.*, defensive/offensive) of each agent.

The definition of the task of sequence imputation is as follows: Let $X = (x_1, x_2, \cdots, x_T)$ denotes a sequence of T observations, where $x_t \in \mathbb{R}^{\mathcal{D}}$ and $\mathcal{D}$ refer to the value of each time step and the dimension of each trajectory, respectively. Typically, some missing data exist in X, represented by a masking sequence $M = (m_1, m_2, \cdots, m_T)$. We set the masking m_t as zero if x_t is missing. The aim of imputation is to replace or complete the missing data with logical or reasonable values in a continuous temporal sequence.

Variational Autoencoders. Here, we review the definition of variational autoencoders (VAE). In order to construct a latent variable model, we define a latent factor generative model by $p_\theta(x, z) = p(z)p_\theta(x|z)$, where θ refers to the parameters of the decoder and $p(z)$ is the prior distribution over latent variables z. The VAE is designed to maximize the log-likelihood of data under such a latent variable model by introducing a variational distribution, namely $q_\phi(z|x)$, which is utilized to encode the input data x and approximate the posterior $p_\theta(z|x)$ of the generative model. ϕ denotes the parameters of the encoder. Hence, we approach a tractable Evidence Lower Bound (ELBO) on likelihood as follows:

$$\begin{aligned} \mathbb{E}_q \log p_\theta(x) &\geq \mathbb{E}_q \log p_\theta(x) - D_{KL}[q_\phi(z|x)||p_\theta(z|x)] \\ &= \mathbb{E}_q \log p_\theta(x|z) - D_{KL}[q_\phi(z|x)||p_\theta(z)]. \end{aligned} \tag{1}$$

4. Proposed Approach

4.1. Overview

Figure 2 illustrates the overall framework of our imitative non-autoregressive modeling approach. We aim at bringing the intuition of imitation learning to the non-autoregressive model and adapt it to our scenario. In our approach, a Recurrent Conditional Variational Autoencoder (RC-VAE) is employed to imitate a knowledgeable *demonstrator*, and a Non-Autoregressive Transformation Model (NART) can be regarded as a *learner*. Through the imitative learning strategy, the RC-VAE is expected to offer efficient supervision and sufficient knowledge to the decoding state of the NART. It is worth pointing out that the RC-VAE is utilized for trajectory prediction, while the NART is employed for missing value imputation. In the following parts, we will describe the details of the *RC-VAE demonstrator* and the *NART learner* in our proposed framework, respectively.

4.2. RC-VAE Demonstrator

We present a recurrent conditional variational autoencoder (RC-VAE) as the demonstrator. The encoder of RC-VAE adopts several conditions as the input, and the decoder has a crucial component, *i.e.*, an imitation learning module. The imitation learning module can emit actions at every time step, which contains necessary temporally contextual information regarded as the supervised guidance during the training process of the NART leaner, and 'action' here can be explained as 'how to correctly decode the hidden variable via the observed data'.

Architecture. In the RC-VAE demonstrator, the main goal is to infer continuous time latent position and velocity trajectories given the observed frames, and learn to model high-order dynamics and capture uncertainties in the time series. Thus, we adopt a variational RNN (VRNN) [7] as our baseline model, which refers to a variational autoencoder (VAE) conditioned on the hidden state of an RNN and can be trained by maximizing the ELBO in Eq. 1. The encoder in our proposed RC-VAE demonstrator is devised in a forward-backward manner. Meanwhile, we also adopt the self-attention layers and feed-forward layers in encoder and decoder that are similar to Transformer [40].

Forward-backward Encoder. Different from the traditional bi-directional RNN, we adopt a forward-backward encoder. The encoder is a two-layer RNN, which maps the observed input and masking sequence into two sets of hidden states, *i.e.*, forward hidden states $H^f = (h_1^f, \cdots, h_T^f)$ and backward hidden states $H^b = (h_1^b, \cdots, h_T^b)$, and the distribution can be denoted as follows:

$$\begin{aligned} q(H^f|I) &= \prod_{t=1}^{T} q(h_t^f | h_{<t}^f, I_{\le t}) \\ q(H^b|I) &= \prod_{t=1}^{T} q(h_t^b | h_{>t}^b, I_{\ge t}), \end{aligned} \tag{2}$$

where h_t^f and h_t^b denote the encoded hidden state from history and the future, respectively, and I refers to the given observation. Then we leverage a two-layer RNN including a forward RNN ('RNN$_f$') and a backward RNN ('RNN$_b$') to parameterize the above distributions:

$$\begin{aligned} q(h_t^f | h_{<t}^f, I_{\le t}) &= \text{RNN}_f(h_{t-1}^f, I_t) \\ q(h_t^b | h_{>t}^b, I_{\ge t}) &= \text{RNN}_b(h_{t+1}^b, I_t). \end{aligned} \tag{3}$$

Training Phase. In the training stage, we need to minimize the Kullback-Leibler (KL) divergence (D_{KL}) in the variational module and the Euclidean distance of the predicted trajectories and the ground truth, *i.e.*, $||Y - \hat{Y}||_2^2$ (where $Y = Y_{\mathcal{P}}^{(t_q, t_v]}$ means the ground truth, $\hat{Y}$ refers to the prediction value, and $\mathcal{P}$ denotes a subset of all agents needed to forecast). Therefore, we construct the loss function as follows:

$$\mathcal{L}_{\text{RC-VAE}} = ||Y - \hat{Y}||_2^2 + \beta D_{KL}(Q||P), \tag{4}$$

where $P(z|X_{\Omega}^{[t_0,t_q]}, \mathcal{O}) = \mathcal{N}(0,1)$ is a prior distribution, $Q(z = \hat{z}|X_{\Omega}^{[t_0,t_q]}, \mathcal{O}) \sim \mathcal{N}(\mu_z, \sum_z)$ is the latent distribution modeled as a normal distribution, and $\hat{z}$ is the sampled latent variable. β is the weighting factor. In our proposed model, we set $\beta = 1$ in all experiments.

Imitation Learning Module. In order to derive our framework in an imitation learning fashion, we incorporate an imitation learning module into the decoder. We define the input of each decoder layer $S^l = \{s_1^l, s_2^l, \cdots, s_T^l\}$ as the observed state, $A^l = \{a_1^l, a_2^l, \cdots, a_T^l\} \in \mathcal{A}$ denotes an action sequence in our imitation learning framework, where l refers to the l-th layer in the decoder, $\mathcal{A}$ indicates a finite action space, and T means the time step. In our imitation learning setting, the action distribution of the RC-VAE demonstrator can then be utilized as training signal to the NART learner.

For the purpose of predicting the action A^l, we need to consider the states S^l, and then map S^l to actions by the policies π^l. Let Π denote a policy class, in which each policy $\pi^l \in \Pi$ makes the decision to generate an action distribution sequence A^l given S^l. Meanwhile, let $\pi^l(s_t^l)$ denote the probability of the decision regarding the current state or environment s_t^l. Then, each action can be calculated as follows:

$$a_t = \arg\max(\pi^l(s_t^l)), \tag{5}$$

where

$$\pi^l(s_t^l) = \text{softmax}(\text{FFN}(s_t^l)). \tag{6}$$

Here FFN refers to the feed-forward layer [40]. In Eq. 5, due to the non-differentiable problem, it is difficult to train the policy with the discrete operation $\arg\max(\cdot)$ in an end-to-end learning framework. To overcome such an issue, we decide to compute the expectation of the action a_t's embedding as the action state:

$$e_t^l = \mathbb{E}_{a_t^l \sim \pi^l(s_t^l)} \xi(a_t^l), \tag{7}$$

where $\xi(a_t^l) \in \mathbb{R}^{\mathcal{K}}$ refers to the embedding of the action a_t^l, and $\mathcal{K}$ is the dimension of embedding vector. Then, the states of the next layer can be calculated depended on the current output state of the previous layer and the emitted expectation of action:

$$s_t^{l+1} = \text{Decoder}(e_t^l + s_t^l). \tag{8}$$

Action Distribution Regularization. In the imitation learning framework, one common challenge is that unsupervised clustering of actions can make the distribution of actions unbalanced. Inspired by [44,45], we adopt a method to increase the utilization space. In order to evaluate the cumulative activation level, we define ρ as the cumulative history of each action category:

$$\rho \leftarrow \alpha \cdot \rho + (1 - \alpha) \sum_{t=1}^{T} \pi(s_t)/T, \tag{9}$$

where the parameter α is set to 0.9 in our experiments. The role of ρ is to redistribute the probability of policy $\pi(s_t)$, making the category assignment more balanced. After that, we can do the re-normalization with $\pi(s_t)$ based on the following:

$$\pi'(s_t) = \frac{\pi(s_t)^2/\rho}{\sum_j \pi(s_t)_j^2/\rho_j}. \tag{10}$$

We draw the regularization objective as a KL divergence between $\pi(s_t)$ and the re-distributed $\pi'(s_t)$,

$$\mathcal{L}_{\pi} = \sum_{t} \pi'(s_t) \log \frac{\pi'(s_t)}{\pi(s_t)}. \tag{11}$$

4.3. NART learner

Inspired by the emerging non-autoregressive translation models [16, 28], we present a non-autoregressive transformation model (NART) for sequence imputation. The structure of NART is almost similar to RC-VAE, while the main difference lies in a non-autoregressive inference strategy.

Non-Autoregressive Inference. After achieving the joint hidden states $H = [H^f, H^b]$ from the encoder, the decoder needs to learn the distribution of the whole temporal sequences $p(X|H)$. As shown in Figure 2, the decoder will identify two observed time steps (*e.g.*, x_1 and x_5), and firstly impute the midpoint missing value (*e.g.*, x_3) at each inference iteration. Then, other missing values can be inferred by such a process repeatedly (*e.g.*, x_2 and x_4). During each time step, the decoder of NART employs a decoding function g to map the obtained hidden states that concatenate the forward states h^f_{t-n} and the backward states h^b_{t+n} to a probability distribution:

$$p(x_t|H) = g(h^f_{t-n}, h^b_{t+n}). \tag{12}$$

Training Objective. Given a series of whole temporal sequence $C = \{X^*\}$, let $G_\theta(X, M)$ refers to the generative process of NART learner with parameter θ, and $p(M)$ means the prior probability over the missing values. Here $\widehat{X}$ and $\widehat{x}$ both denote the generated values. Then the training objective of our proposed NART learner can be formulated as follows:

$$\mathcal{L}_{\text{NART}} = \mathbb{E}_{X^* \sim C, M \sim p(M), \widehat{X} \sim G_\theta(X,M)} [\sum_{t=1}^{T} \|\widehat{x}_t - x_t\|_2^2]. \tag{13}$$

Imitation Learning Objective. To facilitate the imitation learning process in our proposed framework, the RC-VAE demonstrator described in Sec 4.2 is capable of generating action distribution $\pi_{\text{RC-VAE}}$ as the posterior guidance, which is expected to supervise the generation process of the NART learner. Then, the NART learner can emit policy distribution $\pi_{\text{NART}}(S) \in \mathbb{R}^{\mathcal{N}}$ by gradually learning from the RC-VAE demonstrator. Formally, the objective of imitation learning is to minimize the cross-entropy loss of the distribution between the policies generated by the RC-VAE demonstrator and the NART learner:

$$\mathcal{L}_{\text{IL}} = -\mathbb{E}_{\pi_{\text{RC-VAE}}(s_t)} \log \pi_{\text{NART}}(s_t). \tag{14}$$

4.4. Joint Objective Learning

To train the whole framework, we need firstly train the RC-VAE demonstrator by combining the action distribution regularization term in Eq. 11 with the VAE loss in Eq. 4:

$$\mathcal{L}_{\text{demon}} = \mathcal{L}_{\text{RC-VAE}} + \lambda_1 \mathcal{L}_{\pi}. \tag{15}$$

During training of NART learner, we combine the imitation learning term in Eq. 14 with the squared error in Eq. 13:

$$\mathcal{L}_{\text{learn}} = \mathcal{L}_{\text{NART}} + \lambda_2 \mathcal{L}_{\text{IL}}, \tag{16}$$

where λ_1 and λ_2 are hyper-parameters that control the weights of the corresponding loss functions. During the training, we take the randomly sampled trajectories as input for RC-VAE and NART. We first train RC-VAE, and then train NART by freezing RC-VAE's parameters, when NART is guided by RC-VAE in a behavioral cloning way. In this way, the parameters of the imitation module in NART will be updated, to ensure its generated actions under the policy π_{NART} match the RC-VAE demonstrator's actions under $\pi_{\text{RC-VAE}}$. Finally, we minimize the cross-entropy between the distributions of two policies based on Eq. 14.

5. Experimental Results

To substantially validate the proposed model, extensive experiments are conducted on three benchmarks for the tasks of sequence prediction and imputation: 1) Multi-agent trajectories movement from professional basketball matches on **Basketball Tracking Dataset** [10]; 2) Real-world Traffic time series on **PEMS-SF Traffic Dataset** [8]; 3) Billiard ball trajectories derived from a physics engine on **Billiards Ball Trajectory Dataset** [12].

5.1. Experimental Settings

Basketball Tracking Dataset [10] totally contains 95,002 sequences that recorded the trajectories of professional basketball players. Each sequence is transformed from the 3D locations of players and ball into a 2D overhead view of the

Methods	Traffic [8]		Billiards [12]					Basketball [10]					
	P-$L_2 \downarrow$	I-$L_2 \downarrow$	P-$L_2 \downarrow$	I-$L_2 \downarrow$	Sin	Step	Wall	P-$L_2 \downarrow$	I-$L_2 \downarrow$	Path-L	OOB	Step	Path-D
MaskGAN [9]	-	6.02	-	1.830	1.095	15.35	0.100	-	0.427	0.793	4.592	9.622	0.680
KNN [13]	-	4.58	-	5.381	1.469	24.59	0.189	-	0.403	0.921	**0.128**	13.24	0.746
GRUI [29]	-	15.24	-	20.57	1.859	28.19	0.225	-	0.398	1.141	4.703	14.95	0.690
Linear Interpolation	-	15.59	-	19.00	1.121	**0.961**	0.247	-	0.422	0.482	2.997	0.522	0.519
NAOMI [28]	-	3.54	-	0.067	**1.006**	7.239	**0.023**	-	0.423	0.573	1.733	2.565	**0.581**
C-VAE [10]	9.23	-	6.56	-	-	-	-	2.66	-	-	-	-	-
VRNN [7]	8.92	-	5.59	-	-	-	-	2.93	-	-	-	-	-
Vanilla LSTM [20]	19.56	-	15.89	-	-	-	-	10.44	-	-	-	-	-
Social LSTM [2]	12.67	-	9.25	-	-	-	-	5.23	-	-	-	-	-
GT	0.00	0.00	0.00	0.000	1.000	1.588	0.018	0.000	0.000	0.556	0.861	1.982	0.580
Our full model	**5.22**	**2.72**	**3.05**	**0.052**	1.012	6.892	0.030	**1.79**	**0.323**	**0.562**	1.922	**2.155**	0.595
Our model w/o imitation	-	3.66	-	0.065	1.107	7.535	0.159	-	0.412	0.595	2.356	2.356	0.602

Table 1. Performance comparison of our full model, baseline models, and the state-of-the-art approaches on Traffic/Billiards/Basketball Datasets in terms of the sequence prediction and imputation. (P-L_2) and (I-L_2) denote averaged distance L2 loss of prediction and imputation on all datasets, respectively. (Sin), (Step) and (Wall) denote sinuosity, step-change (10^{-3}), and refection to wall regarding imputation on Billiards, respectively. (Path-L), (OOB), (Step), and (Path-D) refer to path length, out-of-bounds rate (10^{-3}), step change (10^{-3}) and path difference on Basketball, respectively. GT means the ground truth value. $\downarrow$ refers to the smaller result is better.

court, which is annotated with 2D coordinates (x, y) of 5 players in 50 time steps with 6.25Hz. Following [10], we split the whole dataset into 107,146 training and 13,845 test sequences. Meanwhile, we create a masking sequence for each trajectory with 40 to 49 missing values following [28].

PEMS-SF Traffic Dataset [8] includes 267 training sequences and 173 testing sequences, which is sampled every 10 minutes overall the day. All data consist of 963 dimensions, of which each dimension represents the freeway occupancy rate recorded from various sensors. In our experiments, we create a masking sequence containing 122 to 140 missing values for each testing data following [28].

Billiards Ball Trajectory Dataset [12] contains 4,000 training and 1,000 test sequences generated by the simulator in [12]. In all generated sequence, the size and density of balls are fixed and uniform, and no friction occurred in trajectories. Additionally, the position and velocity of each ball are randomly initialized and roll out the ball for 200 time steps. We create a masking sequence with 180 to 195 missing values for each trajectory following [28].

Metrics. As for sequence prediction, we calculate the L_2 distance (*P-L_2*) between generated future trajectories and the ground truth for each agent averaged over each time, and we utilize *ft* as the measuring unit. Besides, both the observed trajectory history and the prediction length are set to 4s. While, regarding to sequence imputation, we also measure the L_2 loss between imputed values and ground-truth (*I-L_2*), straightness of the generated trajectories (*Sinuosity*), average step size change (*Step*), average trajectory length in 8 seconds (*Path-L*), average out-of-bound rate (*OOB*), and max-min path difference (*Path-D*).

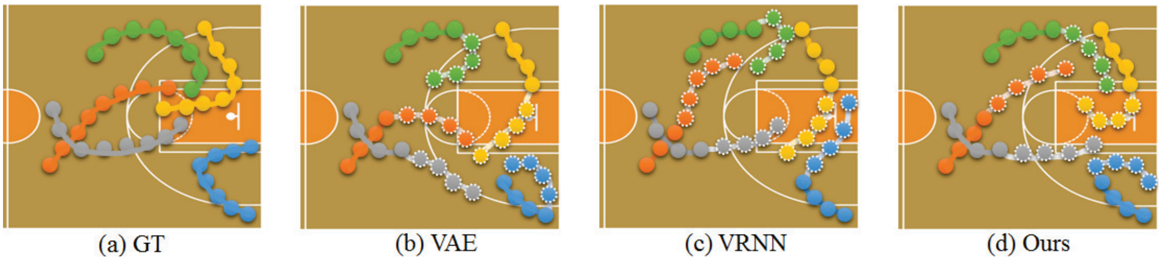
Compared Methods. In experiments, we choose C-VAE [10], VRNN [7], Vanilla LSTM [20], and Social LSTM [2] to compare with our proposed framework for sequence prediction task. While, we select MaskGAN [9], KNN [13], GRUI [29], Linear Interpolation, and NAOMI [28] as baselines for sequence imputation.

5.2. Implementation Details

Our implementation is based on the PyTorch library. We also use PaddlePaddle to implement our method and achieve similar performances. During training, we leverage the ADAM optimizer and set the batch size to 256, initial learning rate to 0.001, and learning rate decay to 0.5 every ten epochs. We randomly sample all the training data to make the experimental results more solid. In our framework, we adopt two-layer LSTMs with 1024-d hidden units and a stack of six self-attention and feed-forward layers as the encoder, and the decoder is also implemented by six self-attention and feed-forward layers following Transformer [40]. As for the experiments on *Basketball Tracking Dataset* [10], we adopt a tree-based role alignment method [36] for trajectory alignment following [10], which is able to minimize reconstruction error and generate an optimal representation of multi-agent trajectories. Additionally, we set both λ_1 and λ_2 as 0.001 in experiments, because we found that the values of $\mathcal{L}_\pi$ and $\mathcal{L}_{\text{IL}}$ were about 10^3 times larger than those of $\mathcal{L}_{\text{RC-VAE}}$ and $\mathcal{L}_{\text{NART}}$ at the beginning of training. Meanwhile, we also found that the performance of our proposed model was not very sensitive to those hyper-parameters.

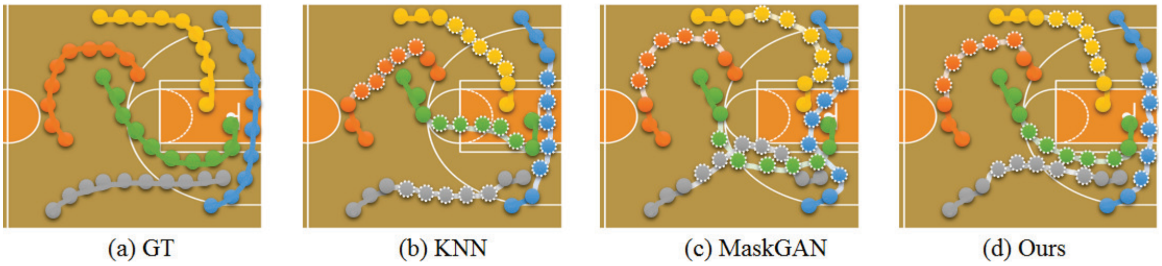
5.3. Results and Analysis

Quantitative Results. Table 1 summarizes the quantitative results of our approach and other baselines in terms of sequence prediction and imputation. From the table, we can



(a) GT (b) VAE (c) VRNN (d) Ours

Figure 3. Visualization examples for basketball trajectories prediction of our proposed method compared with other state-of-the-art approaches. Given a 4s trajectory history for all players, the goal of the task is to predict the next 4s of offense player motion. Circles and lines of different colors refer to the player and the corresponding trajectory, respectively. The light-colored lines and white dotted colored circles denote the predicted trajectories and position of players, respectively. 'GT' means the ground truth.



(a) GT (b) KNN (c) MaskGAN (d) Ours

Figure 4. Visualization of imputed basketball trajectories of our proposed method compared with other state-of-the-art approaches. Given five known observation positions of multi-agent, the goal of the task is to impute the missing value in the sequence. Colored circles denote given discrete observations, and the white dotted colored circle denotes the imputed value, respectively. 'GT' means the ground truth.

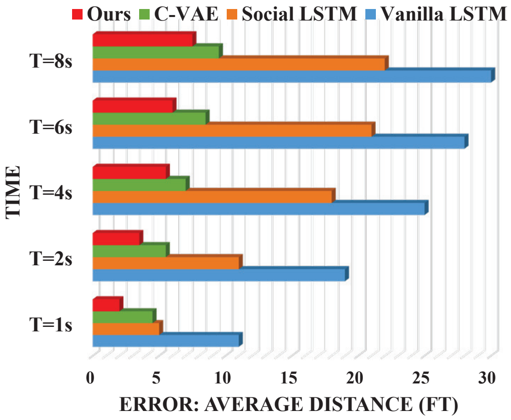


Figure 5. Comparison of our proposed model and other baselines w.r.t prediction errors for 10 agents given 4s observation trajectories on Basketball Tracking Dataset [10].

see our proposed approach manifests the best performance against other methods by a considerable margin in terms of all criteria w.r.t. sequence prediction. For example, the reported P-L_2 of our model amounts to 5.22 on PEMS-SF Traffic Dataset [8] that outperforms other approaches. The significant superiorities of our proposed model compared with other models on Basketball Tracking Dataset [10] also speak to its advantages. Across all the benchmarks, our model achieves the best prediction performance w.r.t P-L_2, which beats other autoregressive models, *e.g.*, C-VAE [10] and VRNN [7], suggesting the introduced recurrent conditional VAE is capable of improving the generated trajectory results. Additionally, our proposed approach achieves the best results on most metrics for the imputation task, such as L_2 loss, *Path-L*, and *Step*. These findings meet our expectations that combining imitative learning and non-autoregressive model into one single framework is a better strategy, and the imitation module plays an important role in bridging the gap between RC-VAE demonstrator and NART learner. However, failing to consider the temporal relevance hinders NAOMI [28] and MaskGAN [9] from achieving satisfactory results. Moreover, we also show that the prediction error increases monotonically with the prediction time horizon in Figure 5. We can observe that our method significantly outperforms the current state-of-the-art approaches. It is worth mentioning that our model obtains the result is four times as good as Social LSTM [2] in each time step, which indicates our proposed forward-backward encoder is better than the conventional RNN structure.

Qualitative Results. We also depict quite a few visualization results to qualitatively verify the generated future trajectories and imputed missing value by our proposed method on Basketball Tracking Dataset [10]. Figure 3 and Figure 4 illustrate the visualized results of trajectory prediction and sequence imputation by our proposed model and other baselines, respectively. It is evident that our proposed model can successfully forecast the future motion trajectories of multiple players and impute the missing value, which is the most consistent with known observations. Conversely,

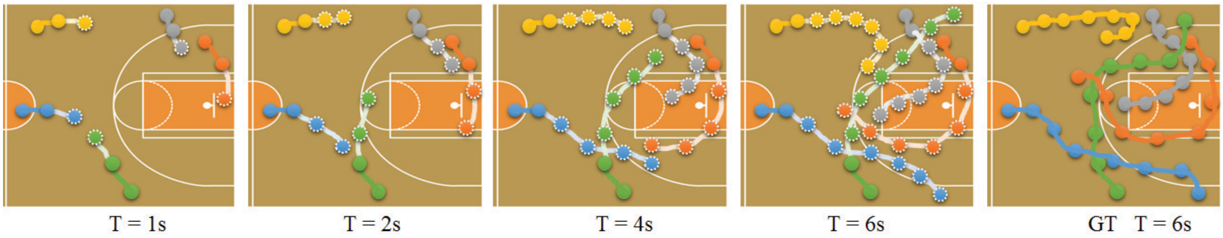


Figure 6. Visualization results of multi-agent's trajectories with time horizon predicted by our proposed method on Basketball Tracking Dataset. Given the previous 2s of each player's motion, we need to generate the future trajectories over horizon time (1/2/4/6 seconds). Each colored circle and line refers to the player and the corresponding trajectory. Light-colored lines and white dotted colored circles denote the predicted trajectories and position of players, respectively. 'GT' means the ground truth.

the results of other baselines still have a large discrepancy to the ground-truth, because these methods disregard the inherent temporal relationship in a sequence. For example, in Figure 4, our model can capture and recover the pattern of the ground-truth time series, while KNN and MaskGAN fail. Because these baselines only learn some averaged behavior instead of continuous temporal dependency and fail to take contextual information into consideration. In addition, to evaluate how long time our proposed model can predict, we provided 4s of multi-agents trajectories history, and then generate the future motion at 1s/2s/4s/6s, as illustrated in Figure 6. We can obviously observe that the generated future trajectories by our model are nearly close to ground-truth in direction, even though some errors still exist. In summary, the qualitative results demonstrate that our proposed framework can generate high-quality future motion trajectories and filling the missing values under various conditions through imitation learning strategy.

5.4. Ablation Studies

Imitation Learning *versus* Non Imitation Learning. We show the performance of our full model and our model without the imitation learning module in Table 1. We can observe that the imitation learning module incorporated into our framework brings an improvement across all the metrics in terms of sequence imputation task. As an example, our framework with the imitation learning module can reduce the prediction and imputation L_2 loss on all three datasets. Our model without imitation learning would susceptible to the unavoidable delayed rewards, and it generates all sequence values, resulting in too large search space. The observation from the comparative results is consistent with our intuition that the temporal relevance learned from RC-VAE demonstrator significantly improves the performance of NART learner.

Action Distribution Regularization. In Subsection 4.2, we propose an action distribution regularization technique in RC-VAE demonstrator to overcome the unbalance problem when searching for actions to supervise NART learner. In experiments, we treat $\pi(s_t)$ with a maximum probability as the selected action during decoding, and the frequency of each action is chosen can be regarded as the used space. We examine the used space on test set of Basketball Tracking [10], and the results are shown in Figure 7. From the figure, we can observe that our proposed action distribution regularization makes the category distribution more balanced and provides more effective guidance for NART learner. In contrast, the model without action distribution regularization leads to most of the space non-utilized, and only few action categories can be used for supervising.

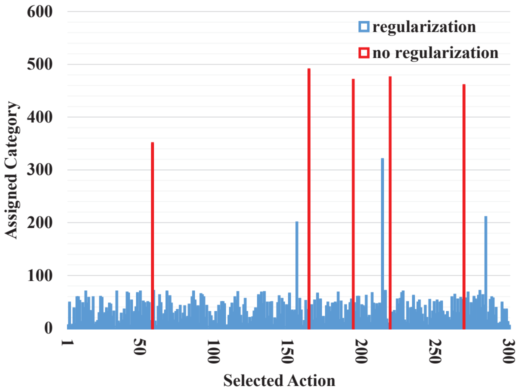


Figure 7. An example of action category assignment distribution on Basketball Tracking Dataset [10]. Action distribution regularization strategy can make the distribution more balanced (in blue), otherwise, it will be overwhelmingly unbalanced (in red).

6. Conclusion

In this paper, we proposed a novel Imitative Non-Autoregressive Modeling method to bridge the performance gap between autoregressive and non-autoregressive models for temporal sequence forecasting and imputation. Our proposed framework leveraged an imitation learning paradigm including two parts, *i.e.* a recurrent conditional variational autoencoder (RC-VAE) demonstrator and a non-autoregressive transformation model (NART) learner. We extensively conducted experiments on three widely-adopted benchmarks, and the promising results demonstrated the effectiveness and superiority of our proposed method by visual examples and quantitative justifications.

References

[1] Edgar Acuna and Caroline Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, clustering, and data mining applications*, pages 639–647. Springer, 2004.

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proc. CVPR*. IEEE, 2016.

[3] Craig F Ansley and Robert Kohn. On the estimation of arima models with missing values. In *Time series analysis of irregularly observed data*, pages 9–37. Springer, 1984.

[4] Lamberto Ballan, Francesco Castaldo, Alexandre Alahi, Francesco Palmieri, and Silvio Savarese. Knowledge transfer for scene-specific motion prediction. In *Proc. ECCV*. Springer, 2016.

[5] 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.

[6] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: bidirectional recurrent imputation for time series. In *Proc. NeurIPS*, 2018.

[7] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Proc. NeurIPS*, 2015.

[8] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[9] William Fedus, Ian Goodfellow, and Andrew M Dai. Maskgan: better text generation via filling in the_. *arXiv preprint arXiv:1801.07736*, 2018.

[10] Panna Felsen, Patrick Lucey, and Sujoy Ganguly. Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In *Proc. ECCV*. Springer, 2018.

[11] Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. In *Proc. ICML*. JMLR, 2016.

[12] Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *arXiv preprint arXiv:1511.07404*, 2015.

[13] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NeurIPS*, 2014.

[15] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

[16] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

[17] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proc. CVPR*. IEEE, 2018.

[18] Irtiza Hasan, Francesco Setti, Theodore Tsesmelis, Alessio Del Bue, Fabio Galasso, and Marco Cristani. Mx-lstm: mixing tracklets and vislets to jointly forecast trajectories and head poses. In *Proc. CVPR*. IEEE, 2018.

[19] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Proc. NeurIPS*, 2016.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Yingfan Huang, HuiKun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proc. ICCV*. IEEE, 2019.

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

[23] Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. Coordinated multi-agent imitation learning. In *Proc. ICML*. JMLR, 2017.

[24] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proc. CVPR*. IEEE, 2017.

[25] Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. In *Proc. NeurIPS*, 2017.

[26] Jindřich Libovickỳ and Jindřich Helcl. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. *arXiv preprint arXiv:1811.04719*, 2018.

[27] Matteo Lisotto, Pasquale Coscia, and Lamberto Ballan. Social and scene-aware trajectory prediction in crowded spaces. In *Proc. ICCVW*. IEEE, 2019.

[28] Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. Naomi: Non-autoregressive multiresolution sequence imputation. In *Proc. NeurIPS*, 2019.

[29] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. In *Proc. NeurIPS*, 2018.

[30] Wei-Chiu Ma, De-An Huang, Namhoon Lee, and Kris M Kitani. Forecasting interactive dynamics of pedestrians with fictitious play. In *Proc. CVPR*. IEEE, 2017.

[31] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*, 2017.

[32] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *Proc. CVPR*. IEEE, 2019.

[33] Mengshi Qi, Jie Qin, Annan Li, Yunhong Wang, Jiebo Luo, and Luc Van Gool. stagnet: An attentive semantic rnn for group activity recognition. In *Proc. ECCV*. Springer, 2018.

[34] Mengshi Qi, Yunhong Wang, Jie Qin, and Annan Li. Kegan: Knowledge embedded generative adversarial networks for semi-supervised scene parsing. In *Proc. CVPR*. IEEE, 2019.

[35] Nicholas Rhinehart, Kris M Kitani, and Paul Vernaza. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proc. ECCV*. Springer, 2018.

[36] Long Sha, Patrick Lucey, Stephan Zheng, Taehwan Kim, Yisong Yue, and Sridha Sridharan. Fine-grained retrieval of sports plays using tree-based alignment of trajectories. *arXiv preprint arXiv:1710.02255*, 2017.

[37] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (TOG)*, 36(4):95, 2017.

[38] Lei Tai, Jingwei Zhang, Ming Liu, and Wolfram Burgard. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In *Proc. ICRA*. IEEE, 2018.

[39] Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, 36(4):93, 2017.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. NeurIPS*, 2017.

[41] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *Proc. ECCV*. Springer, 2016.

[42] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *Proc. ICCV*. IEEE, 2015.

[43] Borui Wang, Ehsan Adeli, Hsu-kuang Chiu, De-An Huang, and Juan Carlos Niebles. Imitation learning for human pose prediction. In *Proc. ICCV*. IEEE, 2019.

[44] Bingzhen Wei, Mingxuan Wang, Hao Zhou, Junyang Lin, and Xu Sun. Imitation learning for non-autoregressive neural machine translation. In *Proc. ACL*, 2019.

[45] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proc. ICML*. JMLR, 2016.

[46] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Gain: Missing data imputation using generative adversarial nets. *arXiv preprint arXiv:1806.02920*, 2018.

[47] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 66(5):1477–1490, 2018.

[48] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision. In *Proc. ICLR*, 2019.

[49] Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.

[50] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *Proc. CVPR*. IEEE, 2019.

[51] Stephan Zheng, Yisong Yue, and Jennifer Hobbs. Generating long-term trajectories using deep hierarchical networks. In *Proc. NeurIPS*, 2016.

[52] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008.