

NAOMI: Non-Autoregressive Multiresolution Sequence Imputation

Yukai Liu¹ Rose Yu² Stephan Zheng^{*3} Eric Zhan¹ Yisong Yue¹

Abstract

Missing value imputation is a fundamental problem in modeling spatiotemporal sequences, from motion tracking to the dynamics of physical systems. In this paper, we take a *non-autoregressive* approach and propose a novel deep generative model: **Non-AutOregressive Multiresolution Imputation (NAOMI)** for imputing long-range spatiotemporal sequences given arbitrary missing patterns. In particular, NAOMI exploits the multiresolution structure of spatiotemporal data to interpolate recursively from coarse to fine-grained resolutions. We further enhance our model with adversarial training using an imitation learning objective. When trained on billiards and basketball trajectories, NAOMI demonstrates significant improvement in imputation accuracy (reducing average prediction error by 60% compared to autoregressive counterparts) and generalization capability for long range trajectories in systems of both deterministic and stochastic dynamics.

1. Introduction

The problem of missing values often arises in real-life sequential data. For example, in motion tracking, trajectories often contain missing data due to targets lying out of the view, object occlusion, trajectories crossing, and the instability of camera motion (Urtasun et al., 2006). Hence, a critical task is missing data imputation which involves filling in missing values with reasonable predictions. Missing data imputation has been studied for decades. Most statistical imputation techniques such as averaging or regression (Rubin, 2004) are either reliant on strong assumptions of the data, or are limited to short-range sequences. In this paper we study how to perform imputation on long-range sequences with arbitrary patterns of missing data.

In recent years, deep generative models and RNNs have

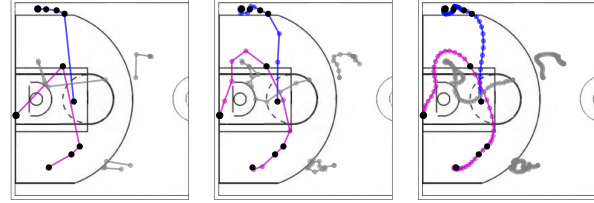


Figure 1. Imputation process of NAOMI in a basketball play given two players (purple and blue) and 5 known observations (black dots). Missing values are imputed recursively from coarse resolution to fine-grained resolution (left to right).

been combined for missing data imputation. A common technique is to append a mask indicating whether or not data is missing, and then train with the masked data (Che et al., 2018). More recently, (Fedus et al., 2018; Yoon et al., 2018a; Luo et al., 2018) propose combining sequence imputation with generative adversarial networks (GAN) (Goodfellow et al., 2014). However, all existing imputation models are autoregressive and impute missing values conditioned on previous data. We find in our experiments that these approaches struggle on long-range sequences with long-term dynamics, as compounding error and covariate shift become catastrophic for autoregressive models.

In this paper, we introduce a novel non-autoregressive sequence imputation method. Instead of conditioning only on previous values, our model learns the distribution of missing data conditioned on both the history and the future. To tackle the challenge of long-term dependencies, we exploit the multiresolution nature of spatiotemporal data, and decomposes the complex sequential dependency into simpler ones at multiple temporal resolutions. Our model, Non-autoregressive Multiresolution Imputation (NAOMI) uses a divide and conquer strategy to recursively fill in the missing values from coarse to fine-grained resolutions. We formalize the learning problem as an imitation learning task, and train the entire model using a differentiable model-based generative adversarial imitation learning algorithm.

In summary, our contributions are as follows:

- We tackle the challenging task of missing value imputation for large-scale spatiotemporal sequences of long-range and long-term dependency.
- We propose NAOMI, a novel deep, non-autoregressive, multiresolution sequence model that recursively im-

¹California Institute of Technology, Pasadena, CA, USA

²Northeastern University, Boston, MA, USA ³Salesforce Research, San Francisco, CA, USA. Correspondence to: Yukai Liu <yukai@caltech.edu>.

^{*}This work was done while the author studied at Caltech.

putes missing values from coarse to fine-grained resolutions with minimal computational overhead.

- When evaluated on billiards and basketball trajectory imputation, our approach outperforms autoregressive counterparts by 60% in accuracy and generates realistic sequences given arbitrary missing patterns.
- Besides imputation, our framework is also effective in forward inference and outperforms state-of-the-art methods without any additional modifications.

2. Related Work

Missing Value Imputation Existing missing value imputation approaches roughly fall into two categories: statistical methods and deep learning methods. Statistical methods either use ad-hoc averaging with mean and/or median values (Acuna & Rodriguez, 2004) or regression models such as ARIMA (Ansley & Kohn, 1984) and MICE (Buuren & Groothuis-Oudshoorn, 2010). Other popular imputation methods include the EM algorithm (Nelwamondo et al., 2007), KNN and matrix completion (Friedman et al., 2001). Despite its rich history, missing value imputation remains elusive as traditional methods are not scalable, often requires strong assumptions on the data, and cannot capture long-term dependencies in sequential data.

Recently, imputation using deep generative models has attracted considerable attention. A common practice is to append a missing indicator mask to the input sequence. For example, M-RNN (Yoon et al., 2018b) proposes a new RNN architecture to account for multivariate dependencies. GRU-D (Che et al., 2018) and BRITS (Cao et al., 2018) modify the RNN cell to capture the decaying influence of missing variables. GAIN (Yoon et al., 2018a) and GRUI (Luo et al., 2018) also use GAN. But GAIN does not model the sequential nature of data, and GRUI does not explicitly take the known observations into account during training. Perhaps the work that most closely related to ours is MaskGAN (Fedus et al., 2018), which makes use of adversarial training and reinforcement learning objective to fill in the missing texts for language modeling. However, all the imputation models are autoregressive, which suffers from compounding error and inconsistency between generated and actual values at observed points.

Non-Autoregressive Modeling Non-autoregressive models have been studied in the context of natural language processing where target words/sentences become independent given the latent embedding and can be predicted non-autoregressively. For instance, Oord et al. (2018) uses Inverse Auto-regressive Flow (Kingma et al., 2016) to map a sequence of independent variables to a target sequence. Gu et al. (2018) introduce a latent fertility model and treats

the input word’s “fertility” as a latent variable. Other works sharing the similar idea include Lee et al. (2018) and Li-
bovický & Helcl (2018). All these works aim to parallelize sequence generation, which can be used to speed up the inference procedure in traditional autoregressive models. Our work is an innovative demonstration of non-autoregressive modeling for sequence imputation tasks. Our model utilizes the chain rule to factorize the joint distribution of sequences without imposing any additional independence assumption.

Behavioral Cloning To explicitly model the sequential dependencies, we formulate our learning problem as an imitation learning task in the nomenclature of reinforcement learning (RL) (Syed & Schapire, 2008; Ziebart et al., 2008). Using GAN in the sequential setting has been explored in Ho & Ermon (2016); Yu et al. (2017). (Ho & Ermon, 2016) propose Generative Adversarial Imitation Learning (GAIL) by using an equivalence between maximum entropy inverse reinforcement learning (IRL) and GANs. Yu et al. (2017) further modify the reward structure and only allow a single reward when the sequence is completed. However, these models only use simple generators, which limit their ability to model long-range sequences.

In order to generate long-range trajectories, Zheng et al. (2016) propose using manually defined macro goals from trajectories as weak labels to train a hierarchical RNN. Zhan et al. (2019) further extends this idea to the multi-agent setting with a hierarchical variational RNN. However, while using macro goals can significantly reduce the search space, reliably obtaining the macro goals from trajectories can be very difficult. We exploit the multiresolution nature of the spatiotemporal data, and aim to learn the hierarchical structure without supervision. Our method bears affinity with other multiresolution generative models such as Progressive GAN (Karras et al., 2018) and multiscale autoregressive density estimation (Reed et al., 2017).

3. Multiresolution Imputation

Let $X = (x_1, x_2, \dots, x_T)$ be a sequence of T observations, where each time step $x_t \in \mathbb{R}^D$. Some of the values in X are missing and the goal is to replace the missing data with reasonable values. We introduce an accurate and efficient solution for missing value imputation.

3.1. Iterative Imputation

Our model, NAOMI, is a deep, non-autoregressive, multiresolution generative model. As depicted in Figure 2 and Algorithm 1, NAOMI has three components: 1) an encoder that learns the representation of the sequence with missing values; 2) a decoder that generates values for missing observations given the hidden representations of the sequence; and 3) a discriminator to distinguish whether the generated

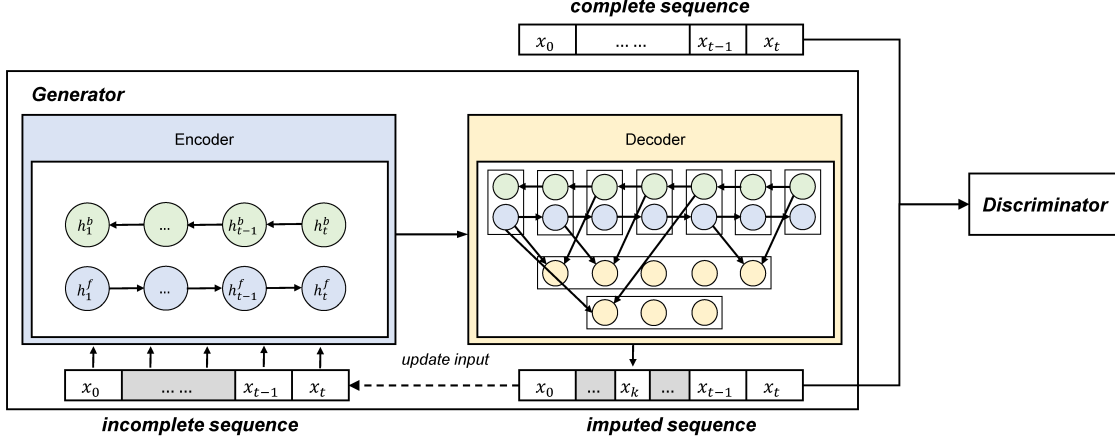


Figure 2. NAOMI architecture. Generator (encoder + decoder) works recursively. At each iteration, the incomplete sequence is first encoded using forward and backward encoders. The multiresolution decoder then predicts one missing value chosen non-autoregressively. This process repeats until all missing values are filled in, and then the imputed sequence is sent to discriminator for training.

sequence is real or fake. The decoder is multiresolutional and operates at different time scales.

The encoder and decoder combined forms a generator G . NAOMI alternates between the encoder and the decoder to impute missing values iteratively. With T_m missing values, we need T_m iterations to impute all missing values. Denote the imputation order as $\mathbf{O} = \{O_1, O_2, \dots, O_{T_m}\}$, by applying the chain rule to factorize the conditional likelihood $P(X|I)$ as a product of conditional probabilities:

$$P(X|I) = P(x_{o_1}, \dots, x_{o_{T_m}} | I) = \prod_{i=1}^{T_m} P(x_{o_i} | I, x_{<o_i}).$$

This factorization allows us to model the conditional likelihood in a tractable manner and without introducing independence assumptions. We begin by describing the encoder for sequences with incomplete data.

3.2. Incomplete Sequence Encoder

To indicate which values in X are missing, we introduce a masking sequence $M = (m_1, m_2, \dots, m_T)$ where $m_t = \mathbf{1}[x_t \text{ is missing}]$ and $\mathbf{1}$ is the indicator function. The concatenated input is then $I = [X, M]$. Our encoder maps I into hidden states $H = [H^f, H^b]$ consisting of forward hidden states $H^f = (h_1^f, \dots, h_T^f)$ and backward hidden states $H^b = (h_1^b, \dots, h_T^b)$. The conditional distribution of the encoder $q(H|I)$ can be decomposed as:

$$\begin{cases} q(H^f|I) = \prod_{t=1}^T q(h_t^f | h_{<t}^f, I_{<t}) \\ q(H^b|I) = \prod_{t=1}^T q(h_t^b | h_{>t}^b, I_{>t}) \end{cases} \quad (1)$$

where h_t^f represents the temporal dependence from the history and h_t^b encodes the dependence on the future.

We can parameterize the above distributions with a forward RNN f_f and a backward RNN f_b . At time step t , f_f encodes the history of observations and imputed values up to t , and f_b encodes the future and missing masks after t :

$$\begin{cases} q(h_t^f | h_{<t}^f, I_{<t}) = f_f(h_{t-1}^f, I_t) \\ q(h_t^b | h_{>t}^b, I_{>t}) = f_b(h_{t+1}^b, I_t) \end{cases} \quad (2)$$

For f_f and f_b , we use GRU cells with ReLU activations. Next we describe our multiresolution imputation decoder.

3.3. Multiresolution Imputation Decoder (MID)

Given the hidden representations H , the decoder learns the distribution of the complete sequence $p(X|H)$. To handle long-term dependencies and avoid error propagation, our decoder uses a *divide and conquer* strategy and imputes values recursively from coarse to fine-grained resolutions. At each iteration, the decoder first identifies two time steps with observed/imputed values as pivots, and imputes a missing value close to their midpoint. One of the pivots is then replaced by the newly imputed point and the process repeats at a higher resolution.

As shown in Figure 3, a multiresolution decoder with R levels is equivalent to a collection of decoders, denoted by $g^{(1)}, \dots, g^{(R)}$, each of which predicts every $n_r = 2^{R-r}$ steps. At resolution r , let the sub-sequence be $X^{(r)} = (x_1, x_{1+n_r}, x_{1+2n_r}, \dots)$. Then $p(X|H)$ can be factorized:

$$p(X|H) = \prod_{r=1}^R p(X^{(r)}|H). \quad (3)$$

The decoder finds two pivots i and j and chooses a time step t with a missing value that is close to the midpoint: $(j - i)/2$. Let r be the smallest resolution that satisfies

Algorithm 1 Non-Autoregressive Multiresolution Imputation

- 1: Initialize generator policy π_θ and discriminator D_ω
- 2: **repeat**
- 3: Sample from expert policy $\tau_E \sim \pi_E$ and mask M
- 4: Compute incomplete sequences $X = \tau_E \odot M$
- 5: Initialize h_t^f, h_t^b using Eqn 2 for $0 \leq t \leq T$
- 6: **while** X contains missing values **do**
- 7: Find the smallest i and j s.t. $m_i = m_j = 1$ and $\exists t$ s.t. $i < t < j$ and $m_t = 0$
- 8: Find the smallest r s.t. $n_r = 2^{R-r} \leq (j - i)/2$
- 9: Select the imputation point $t^* = i + n_r$
- 10: Decode x_{t^*} using Eqn 4, update X, M
- 11: Update h_t^f and h_t^b as follows
- 12: **end while**
- 13: Update generator policy π_θ by backpropagation
- 14: Train discriminator D_ω with π_E and π_θ
- 15: **until** Converge

$n_r \leq (j - i)/2$. The decoder hidden states at time step t are formed by concatenating the forward states $h_{t-n_r}^f$ and the backward states $h_{t+n_r}^b$. Decoder $g^{(r)}$ then maps the hidden states to a probability distribution over the outputs. The masking m_t is updated to 1 after the prediction:

$$p(x_t|H) = g^{(r)}(h_{t-n_r}^f, h_{t+n_r}^b). \quad (4)$$

If the dynamics are deterministic, $g^{(r)}$ directly outputs imputed value. For stochastic environments, we reparameterize the outputs using a Gaussian distribution with diagonal covariance and predict the mean and standard deviation.

Multiresolution as a universal approximator. We provide the theoretical intuition for our multiresolution decoder. Consider an unknown function $f(x_1, x_2, \dots, x_T)$ to be approximated by its multiresolution components at R levels, that is $f(x) \approx f_R(x) = \sum_{r=1}^R g^{(r)}(x)$. Multiresolution approximation is defined as a sequence of functions $g^{(1)}, g^{(2)}, \dots, g^{(R)}$ from a set of nested vector spaces $V_1 \subset V_2 \subset \dots \subset V_r \subset \dots \subset V_R$ that satisfy:

$$g^{(r)} = f(x_1, x_{n_r}, x_{2n_r}, \dots), \quad g^{(r)} \in V_r$$

Based on the wavelet theory (Mallat, 1989), we can decompose any function $f(x)$ using a family of functions obtained by dilating and translating a given scaling function $\phi(x)$, resulting in a discrete wavelet transformation:

$$f(x) = \sum_{r=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} \alpha_{rb} \phi_{rb}(x), \quad \phi_{rb}(x) = \frac{1}{\sqrt{2^r}} \phi\left(\frac{x-b}{2^r}\right)$$

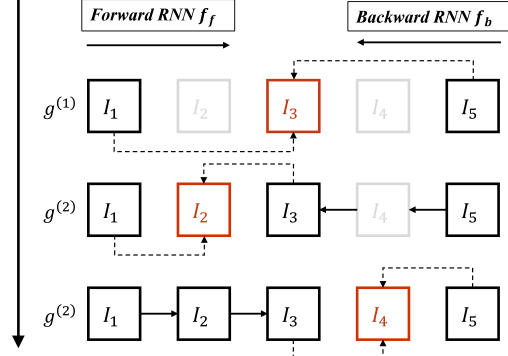


Figure 3. Multiresolution Imputation Decoder in NAOMI. Initially, x_1 and x_5 are observed. We choose the smallest $r = 2$ so that $n_r = 2 \leq 4/2$. Using h_1^f and h_5^b , decoder $g^{(1)}$ imputes x_3 . Then $g^{(2)}$ is used to impute x_2 , and finally x_4 .

where the coefficients $\alpha_{rb} = \int_{-\infty}^{\infty} f(x) \phi_{rb}(x) dx$. Here r denotes the dilation and b defines the translation.

In NAOMI, each decoder approximates the function $g^{(r)}(x) = \sum_b \alpha_{rb} \phi_{rb}(x)$, and we have the recursive formula for neural network approximation: $f_{R+1}(x) = f_R(x) + \sum_{b=-\infty}^{\infty} \alpha_{rb} \phi_{rb}(x)$. Hence, the approximation error at resolution r is bounded by:

$$\epsilon_r = \|f(x) - f_R(x)\| = \left\| \sum_{r=R+1}^{\infty} \sum_{b=-\infty}^{\infty} \alpha_{rb} \phi_{rb}(x) \right\|$$

and becomes progressively smaller as resolution increases.

3.4. Adversarial Imitation Learning

Given the sequential nature of our problem, we cast the imputation task as an imitation learning problem. Given a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ from states to actions, where subsequences $\{s_t = (x_1, x_2, \dots, x_t)\}$ form the state space, and $\{x_t\}$ are the actions. We treat complete sequences as roll outs from an expert policy $\tau_E \sim \pi_E$, where $\tau_E = \{x_1, x_2, \dots, x_T\}$. The generator learns to reconstruct the original sequence given the masked sequence, leading to a learner's policy π_θ . Imitation learning aims to learn a policy π_θ that mimic the expert policy π_E using data.

To quantify the distribution mismatch between reconstructed sequences and training data, we follow the GAIL (Ho & Ermon, 2016) framework. Formally, NAOMI uses the aforementioned generator G parameterized by θ , and a discriminator D parameterized by ω . Our training objective function is:

$$\min_{\theta} \max_{\omega} O(D, G) = \mathbb{E}_{\tau \sim \pi_E} \left[\sum_t D_\omega(s_t, x_{t+1}) \right] + \mathbb{E}_{\tau' \sim \pi_\theta} \left[\sum_t (1 - D_\omega(s_t, x_{t+1})) \right] \quad (5)$$

where the D outputs the probability that state-action pair (s_t, x_{t+1}) comes from data rather than the generator.

One way to optimize the objective in Eqn 5 is to use policy gradients, but this procedure can be expensive and tends to suffer from high variance and sample complexity (Kakade et al., 2003). Instead, we take a model-based approach and assume the environment dynamics are known and differentiable. Hence, we can use the “reparameterization trick” (Kingma & Welling, 2013) to differentiate the generator objective with respect to the policy parameters. Similar ideas have been shown in (Baram et al., 2017; Rhinehart & Kris, 2018) to be more stable and sample-efficient.

4. Experiments

We evaluate NAOMI on the task of missing value imputation in two environments: a billiards physical simulator of a single ball with deterministic dynamics, and a real-world basketball dataset of five player trajectories with stochastic dynamics. We present quantitative and qualitative comparisons with state-of-the-art methods. Lastly, we further investigate our model on the task of forward inference.

Model details. The forward and backward encoders are both 2-layer RNNs with GRU cells. The multiresolution decoder has multiple 2-layer fully-connected neural networks. For the adversarial training, we use a 1-layer RNN with GRU cells as the discriminator. We train on squared loss for billiards and adversarial loss for basketball.

Baselines. We compare NAOMI with a set of baselines that include traditional statistical imputation methods and deep neural network based approaches.

- **KNN:** (Friedman et al., 2001) finds the k nearest sequences in the training set based on known observations. Missing values are imputed using the average of these k -nearest neighbors.
- **Linear:** imputes the missing values using linear interpolation between two known observations.
- **MaskGAN:** (Fedus et al., 2018) uses a single encoder to encode the entire incomplete sequence, a decoder to impute the sequence autoregressively, and a discriminator for adversarial training.
- **GRUI:** (Luo et al., 2018) uses GAN to model the unconditional distribution via a random vector \mathbf{z} . Then it uses **L2** loss to find the best \mathbf{z} based on observed steps. In the original work, complete training sequences are not available and time intervals are not fixed. Here we let the discriminator see the complete training sequence, and we just use the regular GRU cell considering only

fixed time intervals. Hence GRUI may not be perfectly suited for our setting.

- **SingleRes:** has the same encoder structure as NAOMI (a forward and backward encoder), but only has a single resolution decoder to impute missing values. The model is similar to BRITS (Cao et al., 2018), but trained adversarially with a discriminator.

Training. For deterministic settings (e.g. Billiards), we optimize the L2 loss (teacher forcing is applied during pre-training). For stochastic settings (e.g. Basketball), we first pretrain the generator using cross-entropy loss for supervised, and then optimize the generator and discriminator alternatively using the training objective in Eqn 5.

4.1. Imputing Billiards Trajectories

Dataset. We generate 4000 training and 1000 test sequences of Billiards ball trajectories in a rectangular world using the simulator as in (Fragkiadaki et al., 2016). Each ball is initialized with a random position and random velocity and rolled-out for 200 timesteps. All balls have a fixed size and uniform density, and there is no friction in the environment. We generate a masking sequence for each trajectory with 180 to 195 missing values.

Imputation accuracy. The three defining characteristics of the dynamics in this environment are: (1) moving in straight lines; (2) maintaining unchanging speed; and (3) reflecting upon hitting a wall.

To quantify the imputation accuracy of each model, we use four metrics: (1) **L2 loss** between imputed missing values and their ground-truth; (2) *Sinusosity* to measure if the generated trajectory is straight or not; (3) *Average step size change* to measure if the speed of the ball is unchanging; and (4) *Distance between reflection point and the wall* to quantify if the model learns that a ball should bounce against a wall when it collides with one.

Figure 4 compares the model performance using these metrics for imputation accuracy. The average value and 5, 95 percentile values are displayed for each metric. Statistics closer to those from the ground-truth indicate better model performance. NAOMI has the best overall performance across all metrics, followed by our single-resolution baseline model. Note that by design, linear interpolation has an average step size change closest to the ground-truth.

Generated trajectories. We visualize the imputed trajectories from NAOMI and SingleRes in Figure 5. There are 8 known timesteps (black dots), including the starting position. NAOMI can successfully recover the original trajectory whereas SingleRes deviates from the ground-truth. In particular, SingleRes fails to use knowledge of fixed positions in the future to correctly predict the first steps of the

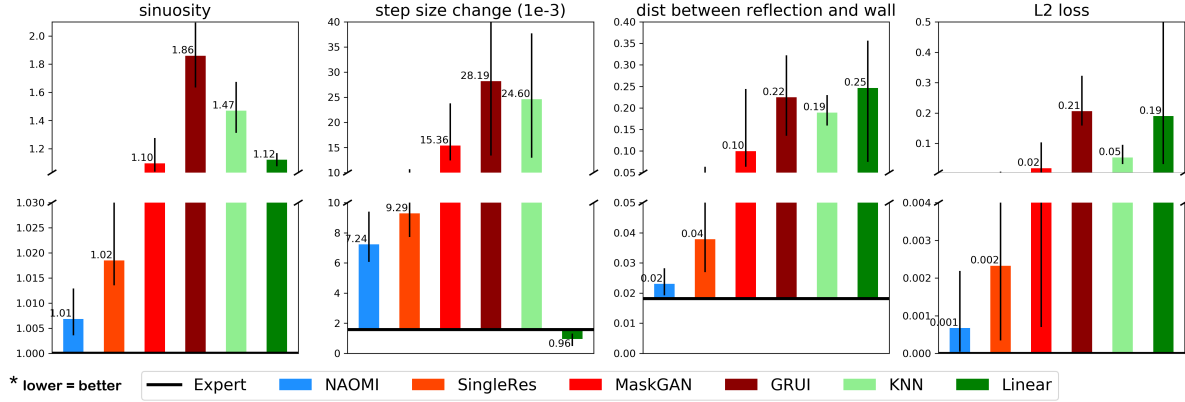


Figure 4. Metrics for imputation accuracy. The average value and 5, 95 percentile values are displayed for each metric. Statistics closer to those from the ground-truth indicate better model performance. NAOMI has the best overall performance across all metrics.

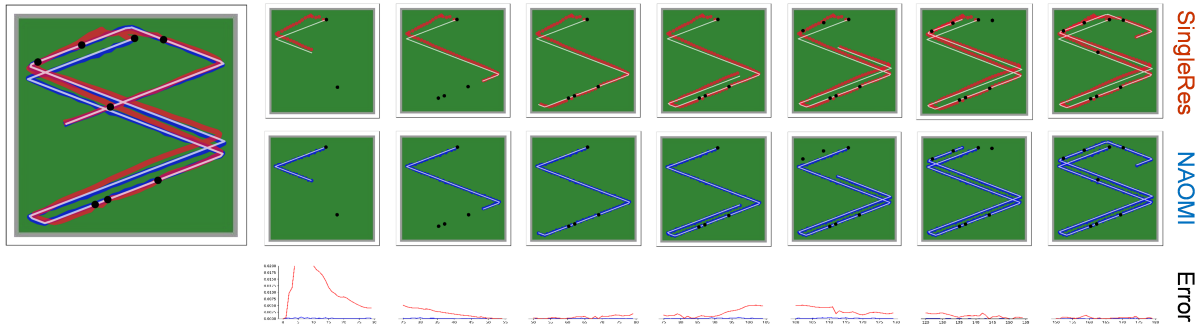


Figure 5. Comparison of imputed billiards trajectories. Blue and red trajectories/curves represent NAOMI and the single-resolution baseline model respectively. White trajectories represent the ground-truth. There are 8 known observations in this example (black dots). NAOMI almost perfectly recovers the ground-truth and achieves lower stepwise L2 loss of missing values than the baseline model (third row). The trajectory from the baseline first incorrectly bounces off the upper wall, which results in curved paths that deviate from the ground-truth as it tries to be consistent with the known observations.

sequence; SingleRes predicts the ball to reflect off the upper wall first instead of the left wall in the ground-truth. As such, SingleRes often has to correct its course to match known observations in the future, leading to curved and unrealistic trajectories. Another deviation from the ground-truth can be seen near the bottom-left corner, where NAOMI produces trajectory paths that are parallel after two reflections, but SingleRes does not.

4.2. Imputing Basketball Plays

Dataset. The basketball dataset contains the trajectories of professional basketball players on offense. Each trajectory contains the (x, y)-coordinates of 5 players for 50 timesteps at 6.25Hz and takes place in the left half-court. In total we have 107,146 training sequences and 13,845 test sequences. We generate a masking sequence for each trajectory with 40 to 49 missing values.

Imputation accuracy. Since the environment is stochastic (basketball players on offense aim to be unpredictable), measuring L2 loss between our model output and the ground-

truth is not necessarily a good indicator of realistic trajectories. Instead, we use the following 5 metrics to quantify how realistic the trajectories are: (1) *Average trajectory length* to measure the typical player movement in 8 seconds; (2) *Average out-of-bound rate* to measure whether the model recognizes court boundaries; (3) *Average step size change* to quantify the relationship between consecutive actions; (4) *Max-Min path diff*; and (5) *Average player distance* to analyze the team coordination. These metrics will serve as our proxy for evaluating imputation accuracy

Figure 6 compares model performance using these metrics for imputation accuracy. The median value and 25, 75 percentile values are displayed for each metric. Statistics closer to the expert data indicate better model performance. NAOMI has the best overall performance across all metrics.

Generated trajectories. We visualize imputed trajectories from all models in Figure 7. NAOMI produces trajectories that are the most consistent with known observations and have the most realistic player velocities and speeds. On the contrary, other baseline models commonly fail in

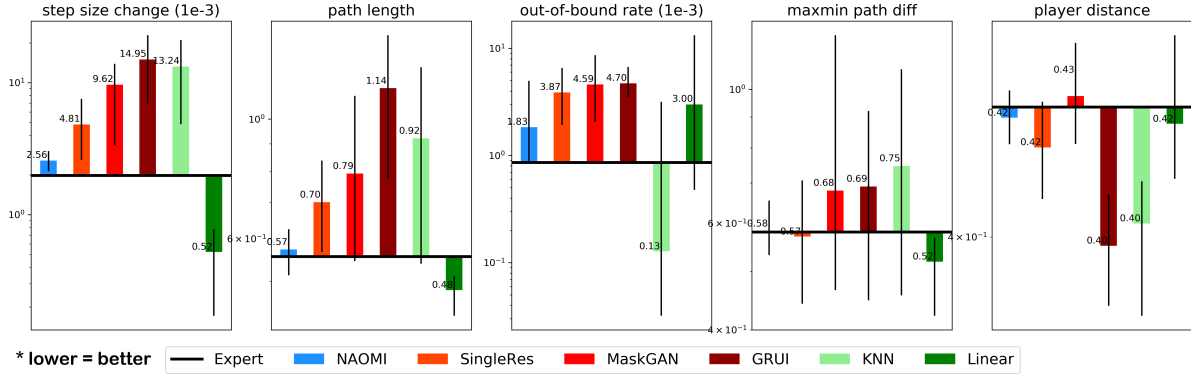


Figure 6. Metrics for imputation accuracy. The median value and 25, 75 percentile values are displayed for each metric. Statistics closer to those from the expert data indicate better model performance. NAOMI has the best overall performance across all metrics.

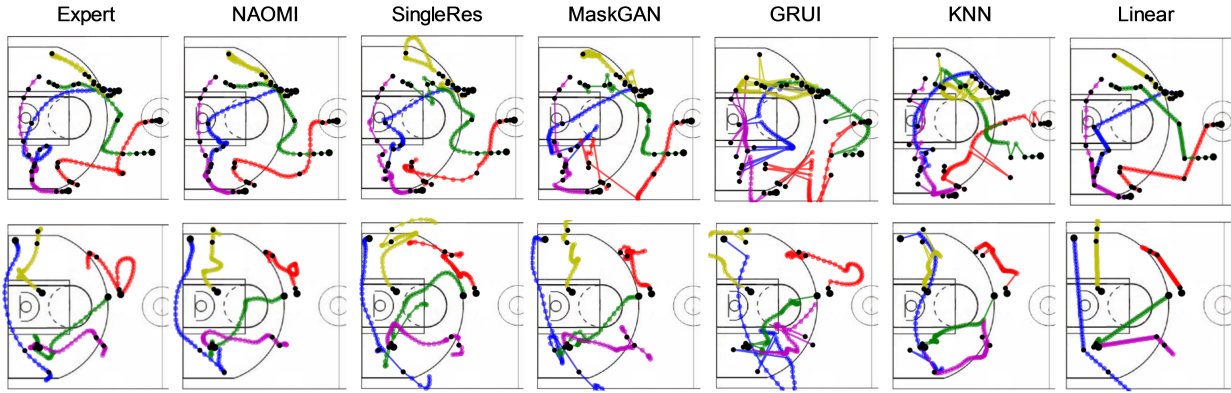


Figure 7. Comparison of imputed basketball trajectories. Black dots represent known observations (10 in first row, 5 in second). Overall, NAOMI produces trajectories that are the most consistent with known observations and have the most realistic player velocities and speeds, whereas other baselines most commonly fail in these regards.

these regards and exhibit problems similar to those observed in our Billiards experiments: KNN generates trajectories with unnatural jumps when there are too many known observations because finding neighbors becomes infeasible; Linear fails to generate trajectories with natural curvature when few observations are known; GRUI fails to generate trajectories consistent with known observations due to mode collapse in the generator when learning the unconditional distribution; and MaskGAN, which consists of a single forward encoder, fails to use known observations in the future and predicts straight lines.

Robustness to percentage of missing values. Figure 8 compares the performance of NAOMI and SingleRes as we increase the number of missing values in the data. Generally speaking, the performance of both models degrade as more missing values are introduced, which makes intuitive sense since imputing more values is a harder task. However, at a certain percentage of missing values, the performance can improve for both models. There is an inherent trade-off between two factors that affect model performance: *Available Information* and *Amount of Constraints*. Observed

information can help models recover the original pattern of the trajectory, but it also serves as constraints that restrict the model output. As we have discussed before, generative models can easily generate trajectories that are consistent with its own historical outputs, but it is harder to be consistent with fixed observations that may deviate from its training distribution. Especially when mode collapse occurs, a single observation not captured by the model can lead to unpredictable trajectory generations.

Learned conditional distribution. Our model learns the conditional distribution $p(X|I)$ of the complete sequence given observations, which we visualize in Figure 9. For a given set of known observations, we use NAOMI to impute missing values with 50 different random seeds and overlay the generated trajectories. We can see that as the number of known observations increases, the uncertainty of the conditional distribution decreases. However, we also observe some mode collapse in our model: the trajectory of the purple player in the right image is not captured in the conditional distribution in the left image.

Table 1. Billiard Forward Inference Metrics Comparison. Better models should have stats that are closer to the expert.

Models	RNN	SingleRes	NAOMI	Expert
Sinuosity	1.054(+5.4%)	1.038(+3.8%)	1.020(+2%)	1.00
Step Change (10^{-3})	11.6(+629%)	9.69(+510%)	10.8(+581%)	1.59
Reflection point dist	0.074(+338%)	0.068(305%)	0.036(+114%)	0.018
L2 Loss (10^{-3})	4.698	4.753	1.682	0.0

Table 2. Basketball Forward Inference Metrics Comparison. Better models should have stats that are closer to the expert.

Models	RNN	RNN + GAN	HVRNN	SingleRes	NAOMI	Expert
Path Length	1.36(+138%)	0.62(+9%)	0.67(+18%)	0.62(+9%)	0.55(-3%)	0.57
OOB Rate (10^{-3})	29.2(+1395%)	4.33(+122%)	7.16(+266%)	3.62(+86%)	3.07(+57%)	1.95
Step size Change (10^{-3})	10.0(+403%)	2.20(+11%)	2.70(+36%)	2.35(+19%)	2.46(+23%)	1.99
Path Difference	1.07(+91%)	0.41(-27%)	0.59(+5%)	0.42(-25%)	0.45(-20%)	0.56
Player Distance	0.450(+7%)	0.402(-4%)	0.416(-2%)	0.412(-3%)	0.422(-0%)	0.424

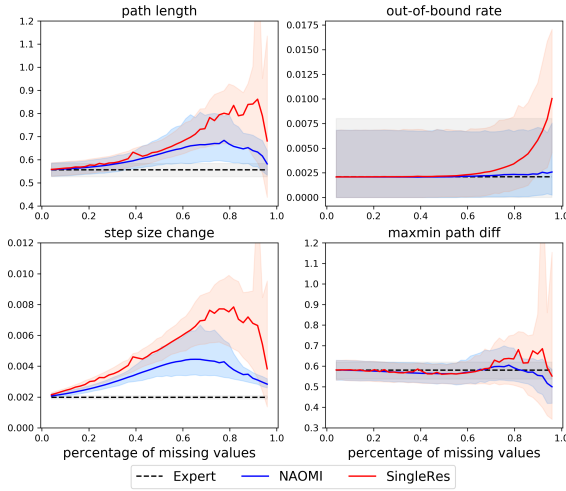


Figure 8. Model performance with increasing percentage of missing values. Statistics closer to the expert indicates better performance. NAOMI performs better than SingleRes for all metrics.

4.3. Imputation as Forward Inference

Imputation reduces to forward inference when all observations, except for a leading sequence, are missing. We show that NAOMI can also be trained to perform forward inference without modifying the model structure. We take a trained imputation model as initialization, and continue training for forward inference by using the masking sequence $m_i = 0, \forall i \geq 1$ (first step is known). We evaluate forward inference performance using the same metrics.

Table 2 shows the quantitative comparison of different models using our metrics. RNN statistics significantly deviate from the ground-truth, but greatly improve with adversarial training. HVRNN (Zhan et al., 2019) uses “macro goals”, and performs reasonably w.r.t macro metrics including average path length and max-min path difference. However, the big step size changes lead to unnatural trajectories. SingleRes has similar performance as the the naive RNN

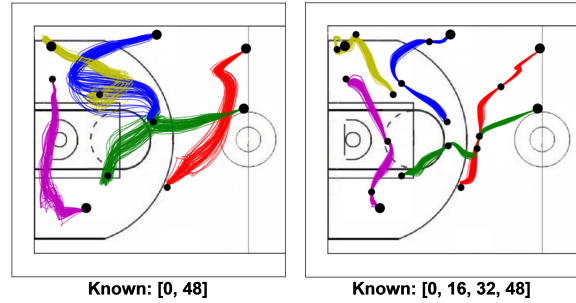


Figure 9. The generated conditional distribution of basketball trajectories given known observations (black dots). As the number of known observations increases, model uncertainty decreases.

+ GAN model, which means our imputation model structure can be used to do forward inference. Finally, the single resolution model does better than NAOMI in terms of the average step size change, but NAOMI has the best performance across all metrics.

Similarly, Table 1 compares forward inference performance in Billiards. NAOMI generates straighter lines and learns the reflection dynamics better than other baselines.

5. Conclusion

We propose a deep generative model NAOMI for imputing missing data in long-range spatiotemporal sequences. NAOMI recursively finds and predicts missing values from coarse to fine-grained resolutions using a non-autoregressive approach. Leveraging multiresolution modeling and adversarial training, NAOMI is able to learn the conditional distribution given very few known observations and achieves superior performances in various experiments of both deterministic and stochastic dynamics. Future work will investigate how to infer the underlying distribution when complete training data is unavailable. The trade-off between partial observations and external constraints is another direction for deep generative imputation models.

References

- Acuna, E. and Rodriguez, C. The treatment of missing values and its effect on classifier accuracy. In *Classification, clustering, and data mining applications*, pp. 639–647. Springer, 2004.
- Ansley, C. F. and Kohn, R. On the estimation of arima models with missing values. In *Time series analysis of irregularly observed data*, pp. 9–37. Springer, 1984.
- Baram, N., Anschel, O., Caspi, I., and Mannor, S. End-to-end differentiable adversarial imitation learning. In *International Conference on Machine Learning*, pp. 390–399, 2017.
- Buuren, S. v. and Groothuis-Oudshoorn, K. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, pp. 1–68, 2010.
- Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. Brits: Bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems 31*, pp. 6776–6786, 2018.
- Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.
- Fedus, W., Goodfellow, I., and Dai, A. Maskgan: Better text generation via filling in the (blank). In *International Conference on Learning Representations (ICLR)*, 2018.
- Fragkiadaki, K., Agrawal, P., Levine, S., and Malik, J. Learning visual predictive models of physics for playing billiards. In *International Conference on Learning Representations (ICLR)*, 2016.
- Friedman, J., Hastie, T., and Tibshirani, R. *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:, 2001.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.
- Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. Non-autoregressive neural machine translation. In *International Conference on Learning Representations (ICLR)*, 2018.
- Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pp. 4565–4573, 2016.
- Kakade, S. M. et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.
- Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.
- Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pp. 4743–4751, 2016.
- Lee, J., Mansimov, E., and Cho, K. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- Libovický, J. and Helcl, J. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- Luo, Y., Cai, X., Zhang, Y., Xu, J., et al. Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems*, pp. 1603–1614, 2018.
- Mallat, S. G. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.
- Nelwamondo, F. V., Mohamed, S., and Marwala, T. Missing data: A comparison of neural network and expectation maximization techniques. *Current Science*, pp. 1514–1521, 2007.
- Oord, A. v. d., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., Driessche, G. v. d., Lockhart, E., Cobo, L. C., Stimberg, F., et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International Conference on Machine Learning*, 2018.
- Reed, S., Oord, A. v. d., Kalchbrenner, N., Colmenarejo, S. G., Wang, Z., Belov, D., and de Freitas, N. Parallel multiscale autoregressive density estimation. In *International Conference on Machine Learning*, 2017.
- Rhinehart, N. and Kris, M. R2p2: A reparameterized push-forward policy for diverse, precise generative path forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 772–788, 2018.
- Rubin, D. B. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

- Syed, U. and Schapire, R. E. A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, pp. 1449–1456, 2008.
- Urtasun, R., Fleet, D. J., and Fua, P. 3d people tracking with gaussian process dynamical models. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pp. 238–245. IEEE, 2006.
- Yoon, J., Jordon, J., and van der Schaar, M. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, 2018a.
- Yoon, J., Zame, W. R., and van der Schaar, M. Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, 2018b.
- Yu, L., Zhang, W., Wang, J., and Yu, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pp. 2852–2858, 2017.
- Zhan, E., Zheng, S., Yue, Y., and Lucey, P. Generating multi-agent trajectories using programmatic weak supervision. In *International Conference on Learning Representations (ICLR)*, 2019.
- Zheng, S., Yue, Y., and Hobbs, J. Generating long-term trajectories using deep hierarchical networks. In *Advances in Neural Information Processing Systems*, pp. 1543–1551, 2016.
- Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.