BlockEcho: Retaining Long-Range Dependencies for Imputing Block-Wise Missing Data

Qiao Han[1] , **Mingqian Li**[1] , **Yao Yang**[1] and **Yiteng Zhai**[1,2*]
[1]Zhejiang Lab
[2]Nanyang Technological University
{hanq, mingqian.li, yangyao, ito}@zhejianglab.com

Abstract

Block-wise missing data poses significant challenges in real-world data imputation tasks. Compared to scattered missing data, block-wise gaps exacerbate adverse effects on subsequent analytic and machine learning tasks, as the lack of local neighboring elements significantly reduces the interpolation capability and predictive power. However, this issue has not received adequate attention. Most SOTA matrix completion methods appeared less effective, primarily due to overreliance on neighboring elements for predictions. We systematically analyze the issue and propose a novel matrix completion method "BlockEcho" for a more comprehensive solution. This method creatively integrates Matrix Factorization (MF) within Generative Adversarial Networks (GAN) to explicitly retain long-distance inter-element relationships in the original matrix. Besides, we incorporate an additional discriminator for GAN, comparing the generator's intermediate progress with pre-trained MF results to constrain high-order feature distributions. Subsequently, we evaluate BlockEcho on public datasets across three domains. Results demonstrate superior performance over both traditional and SOTA methods when imputing block-wise missing data, especially at higher missing rates. The advantage also holds for scattered missing data at high missing rates. We also contribute on the analyses in providing theoretical justification on the optimality and convergence of fusing MF and GAN for missing block data.

1 Introduction

The issue of missing data is prevalent in real-world datasets, stemming from factors such as users' reluctance to share data, variations in feature structures, privacy concerns, and unintended data corruption over time [Scheffer, 2002]. In scenarios involving large-scale and highly correlated missing data, matrix completion becomes essential to impute the unknown entries accurately and efficiently. Accurately and efficiently estimating the unknown elements in a matrix is pivotal for effective dataset analysis and supports subsequent machine learning and operations research tasks. Matrix completion finds wide application in recommender systems, trajectory recovery, phase retrieval, computer vision, genotype imputation and other fields of research [Santos *et al.*, 2019]. Missing data isn't limited to randomly scattered elements; it can also occur in blocks, referred to as block-wise missing data. For example, discrepancies in feature sets provided by different agents can result in non-overlapping blocks of NA values during data aggregation. Moreover, certain regions may experience skipped clinical trials due to technical and financial constraints, while continuous data corruption can arise from malfunctioning monitoring devices over time.

A diverse array of methods have been developed over the years to address the matrix completion problem, spanning traditional techniques like linear interpolation, K-nearest neighbors imputation, Multiple Imputation by Chained Equations (MICE), and ensemble methods such as MissForest [Williams, 2015]. In the new data regime, Generative Adversarial Networks (GANs) have also demonstrated immense potential in accurately estimating high-dimensional data distributions [Goodfellow *et al.*, 2020]. Consequently, recent works have explored GAN architectures for data recovery in the missing data context. However, most prevailing completion techniques display suboptimal effectiveness on block-missing matrices, especially with increasing missing proportions. They predominantly leverage neighboring available elements to predict unknown entries, making them vulnerable to systematic feature gaps. An exception is Matrix Factorization (MF) which implicitly retains inter-sample dependencies across longer ranges [Hastie *et al.*, 2015]. But its linearity assumptions limit capturing complex nonlinear relationships. To unite their complementary strengths and offset limitations, we propose **BlockEcho** - an integrated approach tailored for block-missing data that capitalizes on the strengths of both GAN and MF. It explicitly retains long-range inter-element associations via MF while modeling intrinsic nonlinear patterns through GANs. The key contributions are:

1. We have conducted an in-depth analysis and formally defined the concept of "Block-wise" missing data. Subsequently, we have innovatively introduced a solution to address this challenge, named BlockEcho.
2. We extensively benchmark BlockEcho against SOTA baselines on diverse public datasets for traffic, epidemi-

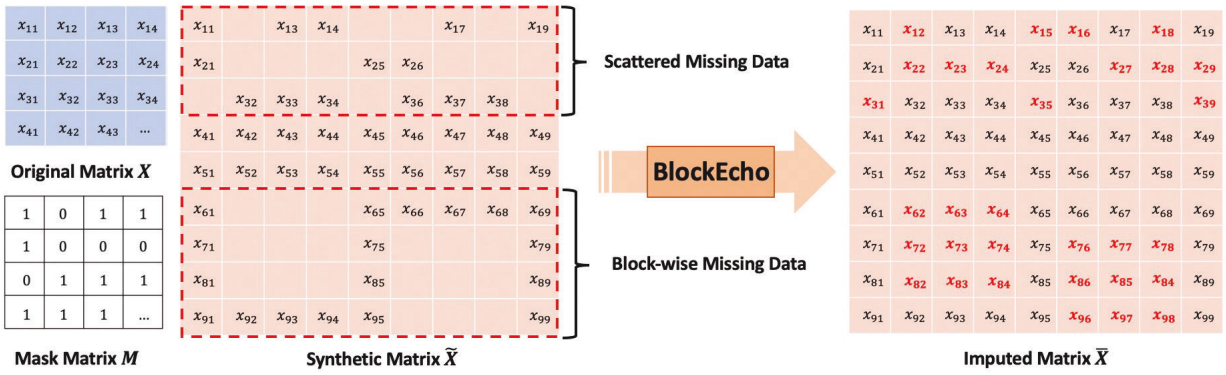


Figure 1: Problem definition

ology, and recommendations. Results demonstrate significant gains in imputation accuracy under both block-missing and high-rate scattered missing regimes. Downstream forecasting tasks further showcase advantages.

3. We provide theoretical analysis into global optimality and convergence for the proposed integrated objective. We particularly justify the synergistic effects of MF encoding long-range dependencies and GANs locally adapting complex distributions. Rigorous proofs supplement empirical observations.

The rest of this paper is structured as follows. In Section 2 we provide a concise literature review in missing data imputation studies. Section 3 defines the problem. We elaborate our model BlockEcho in Section 4. Section 5 provides a theoretical discussion. Section 6 details our experiments and results. We conclude the paper in Section 7.

2 Related Work

2.1 Data Imputation Approaches

Broadly, prevailing techniques for missing data imputation fall under two categories – discriminative or generative. Discriminative methods directly estimate the conditional distribution of missing values given observed entries. This includes MICE [Buuren and Oudshoorn, 2000], MissForest [Stekhoven and Bühlmann, 2011] and Matrix Factorization [Hastie *et al.*, 2015] [Yu *et al.*, 2016]. Generative approaches instead model the complete joint data distribution and draw imputes from conditionals on observed parts. Traditionally, expectation maximization algorithms iteratively refining estimates were common [Gold and Bentler, 2000]. Recently, advances in deep generative modeling have spawned powerful imputation paradigms. Notably, J. Yoon et al. [Yoon *et al.*, 2018] pioneered Generative Adversarial Networks (GANs) for missing data imputation through the GAIN framework in 2018. Instead of assuming explicit parametric forms, GANs can flexibly approximate arbitrary data distributions. Consequently, GAN-based solutions have risen as SOTA, significantly improving over pre-GAN techniques.

2.2 GAN-Based Methods

Many works have since built upon GAIN, either enhancing its convergence like SGAIN (slim Wasserstein GAIN) [Neves *et al.*, 2021], integrating auxiliary models as in GAMIN (Generative adversarial multiple imputation network) [Yoon and Sull, 2020], or exploiting complementary data modalities like images via CollaGAN (Collaborative GAN for missing image data imputation) [Lee *et al.*, 2019]. Some explore sophisticated relationships within data beyond distributions, including: 1) Dual generative modeling of values and high-dimensional feature abstractions in MisGAN [Li *et al.*, 2019], 2) time-series self-attention mechanisms in [Zhang *et al.*, 2021] to capture sensor inter-dependencies, 3) Pretraining classifier-guided generative models as PC-GAIN [Wang *et al.*, 2021], 4) Distribution-centered losses like STGAN [Yuan *et al.*, 2022], and 5) Multimodal spatio-temporal modeling via GAN+RNN+GCN [Kong *et al.*, 2023]. ANODE-GAN [Chang *et al.*, 2023] also augments GANs with auxiliary variational autoencoders.

Nonetheless, prevailing GAN imputation approaches overlook systematic data deficiencies like block-missing data. Mostly relying on local neighborhoods, they remain less effective as gaps exacerbate, especially at higher missing rates. Our proposed BlockEcho framework aims to address this limitation by uniquely blending GANs with matrix factorization.

3 Problem Definition

Consider an original data matrix $X \in \mathbb{R}^{m*n}$. $M \in \{0,1\}^{m*n}$ is set as a binary mask matrix codifying missingness, where $M_{ij} = 0$ denotes element X_{ij} as unobserved. Imposing M on X gives the incomplete matrix:

$$\tilde{X}_{ij} = \begin{cases} X_{ij} & if M_{ij} = 1 \\ NaN & if M_{ij} = 0 \end{cases} \quad (1)$$

We refer to X as the original matrix, M as the mask matrix, and $\tilde{X}$ as the missing data matrix. The data imputation problem aims to obtain an estimation $\hat{X}$ of X given $\tilde{X}$ and M, with the objective of minimizing the discrepancy between the imputed data and the original data. Our final output is the imputed matrix $\bar{X}$ defined as

$$\bar{X} := \tilde{X} \odot M + \hat{X} \odot (1 - M) \quad (2)$$

where $\odot$ denotes element-wise multiplication.

For any given coordinates (i_l, j_l), $\exists\ i_u \geq i_l + 3$ and $j_u \geq j_l + 3$, $\forall\ i_l \leq i \leq i_u$ and $j_l \leq j \leq j_u$, $M_{ij} = 0$,

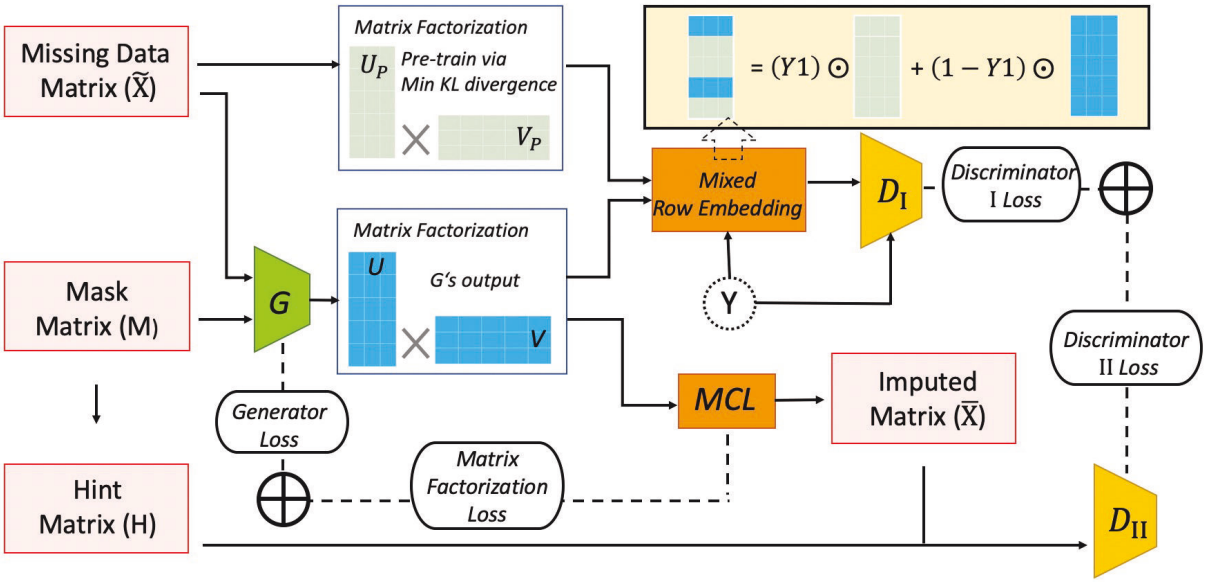


Figure 2: The architecture of BlockEcho

Ori-Data	Scattered	Tri-Blocks	Uni-Block
0.0395	0.0961	0.1132	0.1249

Table 1: WMAPE of prediction task in different missing type

then we define this area as exhibiting a local block-wise missing pattern. Following the definition, the missing data in datasets can manifest as uni-block-wise, multiple-separate-block-wise, scattered, or as a mixed combination thereof.

We conducted a straightforward experiment to understand the harm of missing data. We employed a random forest algorithm to forecast the next timestamp's traffic flow for dataset PE-BAY (outlined in Section 7), without any feature engineering but normalization. Each category of missing data exhibited a 60% missing rate. Our experimental findings, as presented in Table 1, highlighted the notably adverse impact of a substantial, uniformly absent data block on the accuracy of predictions. Henceforth, unless explicitly specified, "block-wise missing" denotes the uni-block.

4 The Architecture of "BlockEcho"

In this section, we will elaborate on our integrated model, elucidating how the MF and GAN components synergize to enhance overall performance. The overarching model architecture of BlockEcho is depicted in Figure 2.

4.1 Matrix Factorization (MF) and Pre-training

Within the MF component, the missing data matrix is decomposed into two smaller matrices, denoted as the imputed row embedding matrix U and the imputed column embedding matrix V. Their dimensions are $m * h$ (adjustable to mini_batch_size$*h$ during training) and $h * n$, where h is a hyperparameter. In essence, h serves as an indicator of the original matrix's level of freedom, with smaller values reflecting a higher degree of interdependency among the data in the original matrix.

As illustrated in Figure 2, there are two MF components in BlockEcho. One is deployed during the pre-training stage, breaking down the matrix into U_p and V_p to extract long-range relationship features, which are then used to supervise model training in a transformed space through a discriminator D_I. The other is directly involved in neural network training and result generation: the imputed row embedding matrix U is produced by the generator, while the imputed column embedding matrix V is directly calculated through backpropagation. Subsequently, these matrices undergo a matrix completion layer (MCL) to match the known parts of the missing data matrix. Rather than employing direct multiplication, MCL incorporates a fully connected neural network layer after multiplication. This technique is employed to introduce nonlinear relationships while upholding the inherent advantage of low constraints on the degree of freedom in the matrix factorization method. Both MF components are oriented towards minimizing the Kullback-Leibler divergence of known elements, thereby establishing the MF objective as

$$\min \sum_{i,j}((X \odot M)_{ij} log(\frac{(X \odot M)_{ij}}{(MCL(UV) \odot M)_{ij}}) - (X \odot M)_{ij} + (MCL(UV) \odot M)_{ij}) \quad (3)$$

Noted that KL divergence is used as the loss function and it demonstrates the best performance experimentally, since all three of our datasets are non-negative and closer to a Poisson distribution. Alternatively, for datasets containing negative values, we can chose the MSE loss function as a substitute.

4.2 Generative Adversarial Nets (GANs)

The GAN framework consists of a generator G and two discriminators D_I, D_{II} as denoted in Figure 2. The generator G

takes $\tilde{X}$ with 0 imputed, M, and a noise matrix Z with the same size as U as the inputs, and U as the output. Next, MCL takes U as input and outputs the estimation $\hat{X}$. With $\hat{X}$, $\tilde{X}$ and M, we build the imputed matrix $\bar{X}$ following Equation (2). Given a vector of length h, the discriminator D_I will determine whether it originates from pre-trained matrix factorization or from the generator. To do this, we randomly generate $Y \in \{0,1\}^{m*1}$ as the index vector and build a mixed-row matrix U_D combining U_p and U:

$$U_D = (YI) \odot U_p + (1 - YI) \odot U \tag{4}$$

where $I \in \{1\}^{1*h}$. The discriminator D_I will output a result of the same size as Y and the objective is

$$\min_G \max_{D_I} \; Y^T log D_I(U_D) + (1-Y)^T log(1 - D_I(U_D)) \tag{5}$$

The discriminator D_{II} is designed to discern whether the elements in $\bar{X}$ is from $\tilde{X}$ (real) or $\hat{X}$ (fake). This differs from a standard GAN model which aims to differentiate inputs ranging from entirely true to entirely false. In addition to $\bar{X}$, D_{II} incorporates the hint matrix H as input. Hint matrix, proposed in GAIN method [Yoon *et al.*, 2018], modifies the elements of M to 1/2 at a specified rate. This modification facilitates D_{II} to converge more swiftly and accurately by furnishing "enough" information regarding M. The objective of this component is

$$\begin{aligned} \min_G \max_{D_{II}} \sum_{i,j} & (M \odot log D_{II}(\bar{X}, H) \\ & + (1 - M) \odot log(1 - D_{II}(\bar{X}, H)))_{ij} \end{aligned} \tag{6}$$

In particular, the MF component serves as a structured constraint and guidance for the GAN component (analogous to the relationship between CNN and MLP), enabling the entire network to explicitly preserve long-range relationships (similar to CNN's explicit preservation of proximity relationships). This fundamental characteristic underpins the superior performance of our model compared to baselines on datasets with missing blocks or high missing rates.

4.3 Objective

According to (3), (5) and (6), the objective of the whole multi-loss model is

$$\begin{aligned} \min_G \max_D \; & (1-\alpha)[(1-Y)^T log(1 - D_I(U_D)) \\ & + Y^T log D_I(U_D) + \sum_{i,j}(M \odot log D_{II}(\bar{X}, H) \\ & + (1 - M) \odot log(1 - D_{II}(\bar{X}, H)))_{ij}] \\ & + \alpha[\sum_{i,j}((X \odot M)_{ij} log(\frac{(X \odot M)_{ij}}{(G(\tilde{X}, M, Z)V \odot M)_{ij}}) \\ & - (X \odot M)_{ij} + (G(\tilde{X}, M, Z)V \odot M)_{ij})] \\ with \; & U_D = (YI) \odot U_p + (1 - YI) \odot G(\tilde{X}, M, Z), \\ & \bar{X} = \tilde{X} \odot M + MCL(G(\tilde{X}, M, Z)V) \odot (1 - M) \end{aligned} \tag{7}$$

Similar to a standard GAN, we alternate between training the generator and discriminators separately. The MF loss (3) is only connected to the generator and will be utilized for multi-loss training alongside the generator's objective.

5 Theoretical Discussion

5.1 Global Optimality

Without loss of generality, we assume the elements of X as $X_{ij} = f(i,j) + s_{ij}$. Here, f represents the bias effect of coordinates on X. The greater the internal correlation of X's elements, the stronger regularity f shows. $\{s_{ij} : i \in [0, m-1] \, j \in 0, n-1\}$ are a series of random variables which follow independent and identical distribution p_d. We can define $\hat{X}$ in same way, i.e., $\hat{X}_{ij} = \hat{f}(i,j) + \hat{s}_{ij}$, where $\hat{s}$'s probability distribution is p_g. In real-world data, whether an element is masked or not may be affected by the value of the element itself. It also has different effects on the corresponding elements of $\hat{X}$ in generative model. So we need to discuss the probability distribution under different mask conditions. We assume that the distributions of s corresponding to the unmasked elements in X and $\hat{X}$ are $p_{d|m=1}$ and $p_{g|m=1}$, and to other masked elements are $p_{d|m=0}$ and $p_{g|m=0}$. The objective of this problem can be written as

$$\hat{f} \to f \;\; and \;\; p_{g|m=0} \to p_{d|m=0} \tag{8}$$

Throughout this section, we assume that the binary mask matrix M is independent of the original matrix X. Then we can obtain $p_{d|m=0} = p_{d|m=1} = p_d$, and (8) can be transformed into

$$\hat{f} \to f \;\; and \;\; p_{g|m=0} \to p_{d|m=1}, \tag{9}$$

or

$$\hat{f} \to f \;\; and \;\; p_{g|m=1} \to p_{d|m=1} \;\; and \;\; p_{g|m=0} \to p_{g|m=1}. \tag{10}$$

Obviously (8), (9) and (10) are equivalent.

Below we will prove that the theoretical global optimal of (6) is the same as that in (9). The loss function for standard GAN is

$$E_{x \sim p_d}[log D(x)] + E_{z \sim p_g}[log(1 - D(G(z)))] \tag{11}$$

For this question, it can be re-written as:

$$\begin{aligned} & E_{\bar{S} \sim p_{d|m=1}}[log D(\bar{X}, H)] + E_{\bar{S} \sim p_{g|m=0}}[log(1 - D(\bar{X}, H))] \\ & = \int_x p_{d|m=1}(x - f) log D(x, H) \\ & + p_{g|m=0}(x - \hat{f}) log(1 - D(x, H)) \mathrm{d}x \end{aligned} \tag{12}$$

For any $a, b > 0$, the function $alogx + blog(1-x)$ reaches its minimum if and only if $x = \frac{a}{a+b}$. So if G is fixed, the optimal discriminator D is $\frac{p_{d|m=1}(x-f)}{p_{d|m=1}(x-f) + p_{g|m=0}(x-\hat{f})}$. Substituting the optimal discriminator into (12), we can obtain

$$\begin{aligned} & E_{\bar{S} \sim p_{d|m=1}}[log D(\bar{X}, H)] + E_{\bar{S} \sim p_{g|m=0}}[log(1 - D(\bar{X}, H))] \\ & = KL(p_{d|m=1}(x - f) \| \frac{1}{2}(p_{d|m=1}(x - f) + p_{g|m=0}(x - \hat{f}))) \\ & + KL(p_{g|m=0}(x - \hat{f}) \| \frac{1}{2}(p_{d|m=1}(x - f) + p_{g|m=0}(x - \hat{f}))) \\ & - log(4) \end{aligned} \tag{13}$$

We recognize that (12) is essentially optimizing Kullback-Leibler divergence. It will achieve its global minimum $-log(4)$ if and only if $\hat{f} = f$ & $p_{g|m=0} = p_{d|m=1}$, which is the same as (9).

Next, we will calculate the theoretical global optimal of (3). To do this, we initially introduce the entropy of the variable, $Ent(A)$, a measure of the disorder of A. According to the definitions of entropy and mutual information, we can find that

$$\begin{aligned} Ent(\hat{X}) =& Ent(\hat{X} \odot M + \hat{X} \odot (1 - M)) \\ =& Ent(\hat{X} \odot M) + Ent(\hat{X} \odot (1 - M)) \\ & - I(\hat{X} \odot M, \hat{X} \odot (1 - M)) \end{aligned} \tag{14}$$

where I denotes mutual information. Noted that $I(\hat{X} \odot M, \hat{X} \odot (1-M)) \leq Ent(\hat{X} \odot (1-M))$ and the equality holds when $p_{g|m=0} = p_{g|m=1}$, which means that if $p_{g|m=1}$ is fixed, the global optimal of $min(Ent(\hat{X}))$ is $p_{g|m=0} = p_{g|m=1}$. Therefore, (10) is equivalent to a multi-objective optimization problem

$$min\ [D(X \odot M || \hat{X} \odot M),\ Ent(\hat{X})] \tag{15}$$

Where $D(\hat{X} \odot M || X \odot M)$ is the primary objective, which can be Euclidean distance or Kullback-Leibler divergence. Its theoretical optimal solution is $\hat{f} = f$ & $p_{g|m=1} = p_{d|m=1}$ under the assumption that $\hat{X}$ could be fitted by any functions (actually not). $Ent(\hat{X})$ is the second objective. We assume the global optimality of (15) is $p_{g|m=0} = p_{g|m=1} = p_{d|m=1} = p^*$, in which case $Ent(\hat{X}) = Ent^*(\hat{X})$. Thus, we have shown that problem (10) is equivalent to the multi-objective optimization problem (15).

Actually, it is very difficult to quantify $Ent(\hat{X})$ in (15) in practice. Matrix factorization is essentially a constraint on entropy in the form of the network structure. From this premise, we target at solving (15) approximately, and the optimization problem for the matrix factorization can be written as:

$$\begin{aligned} min\ \ & D(X \odot M || \hat{X} \odot M) \\ s.t.\ \ & Ent(\hat{X}) < ent \end{aligned} \tag{16}$$

Where ent is a hyperparameter decided by the network structure (mainly h mentioned in section 4.1). We proceed with the following two cases: 1. $ent > Ent^*(\hat{X})$. In this case, the optimal solution of $\hat{f} = f$ & $p_{g|m=1} \rightarrow p_{d|m=1}$ can be reached theoretically, but it lacks a sufficient solution for $p_{g|m=0} \rightarrow p_{g|m=1}$. It corresponds to overfitting in general machine learning models. 2. $ent < Ent^*(\hat{X})$. In this case, $\hat{x}$ cannot be fully represented by the model due to its low degree of freedom limited by the tighter constraint on entropy, which hinders $\hat{x}$ from reaching the optimal of both $\hat{f} = f$ & $p_{g|m=1} \rightarrow p_{d|m=1}$ and $p_{g|m=0} \rightarrow p_{g|m=1}$. It corresponds to underfitting in general machine learning models. In either case, the theoretical global optimal of MF, i.e., problem (16), is the approximate solution of (15).

5.2 Convergence

In the previous subsection, we demonstrated that loss (3) has a similar but relatively inferior theoretical optimal solution compared to loss (6), even when we introduce non-linear transformations to MF through neural networks. However, it is anticipated intuitively (also be verified by ablation experiments in subsection 6.3), that the long-range dependencies introduced by MF will exert a significant influence on the outcomes of block-missing data imputation. This is due to the inherent challenge faced by models with a higher degree of freedom in effectively converging to fit long-range dependencies, particularly in the case of GAN models. The convergence capacity of GANs has posed a substantial challenge since its inception. In 2017, L Mescheder et al. [Mescheder *et al.*, 2017] postulated that poor convergence is attributed to the eigenvalues of the Jacobian matrix of the gradient vector field having a zero real part and an excessively large imaginary part. Consequently, strategies such as gradient penalty [Gao *et al.*, 2020] and spectral normalization [Miyato *et al.*, 2018] have been adopted to alleviate this limitation. In tackling the specific matrix completion problem at hand, we leverage (3), the MF loss function, and (5), which enables the generator to more accurately emulate MF, to effectively steer the convergence of (6).

At last of the section, we will prove the convergence of loss (3), indicating that $D(\hat{X} \odot M || X \odot M)$ is nonincreasing under certain update rules.

Let F denote this expression. Let u denote the set of parameters of the model, and u^t the set of parameters upon update at step t. We introduce an auxiliary function $G(u, u^t)$ for $F(u)$ satisfying $G(u, u) = F(u)$ and $G(u, u^t) \geq F(u)$.Consequently, we can obtain $F(u^{t+1}) \leq F(u^t)$ when $G(u^{t+1}, u^t) \leq G(u^t, u^t)$, then F is nonincreasing under the update $u^{t+1} = argmin_u G(u, u^t)$.

In our paper, we choose Kullback-Leibler divergence as the loss (3) because the datasets are closer to Poisson distributions. In 2001, Daniel et al.[Lee and Seung, 2000] proposed the general auxiliary function for divergence. For this problem, we can make $F(u)$ equal to loss (3) as $\hat{X}_{i,j} := \sum_a u_{ia} v_{aj}$ and $\hat{X}^t_{i,j} := \sum_a u^t_{ia} v_{aj}$, and define $G(u, u^t)$ as

$$\begin{aligned} G(u, u^t) := & \sum_{i,j} (((X \odot M)_{ij} log(X \odot M)_{ij} \\ & - (X \odot M)_{ij} + (\hat{X} \odot M)_{ij}) \\ & - \sum_a (X \odot M)_{ij} \frac{u^t_{ia} v_{aj}}{\hat{X}^t_{i,j}} (log u_{ia} v_{aj} - log \frac{u^t_{ia} v_{aj}}{(\hat{X} \odot M)^t_{ij}})) \end{aligned} \tag{17}$$

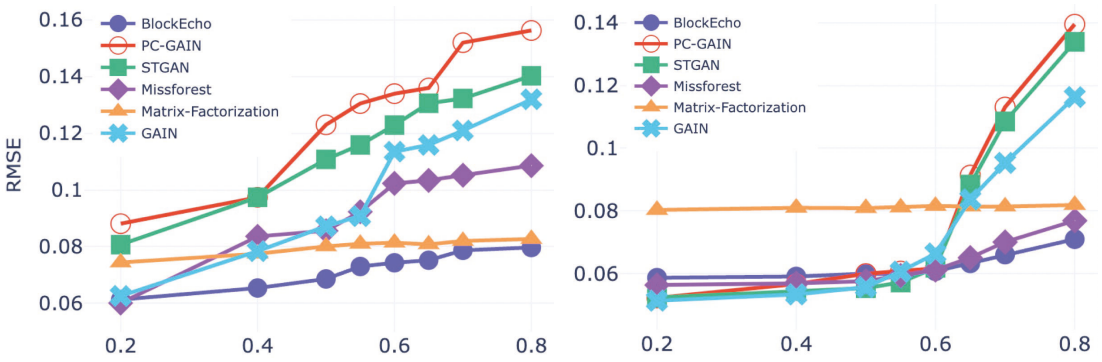
It is straightforward to verify that G satisfies $G(u, u) = F(u)$ and $G(u, u^t) >= F(u)$, and that $F(u^{t+1}) \leq G(u^{t+1}, u^t) \leq G(u^t, u^t) \leq F(u^t)$. Hence, F is nonincreasing under certain update rules and the convergence of loss (3) is proved.

6 Experiments and Evaluation

In this section, we rigorously test the performance of our data imputation method, BlockEcho, through a series of four care-

	Method	ME-LA	PE-BAY	Cov-ca	Cov-de	Movie
Block-wise Missing Data	MissForest	0.2166	0.1037	0.1106	0.1589	–
	Matrix Factorization	0.1637	0.0814	0.0645	0.0802	–
	GAIN	0.2020	0.1135	0.0642	0.0811	–
	PC-GAIN	0.2169	0.1340	0.1055	0.1166	–
	STGAN	0.2057	0.1229	0.0757	0.0968	–
	BlockEcho(ours)	**0.1561**	**0.0743**	**0.0495**	**0.0784**	–
Scattered Missing Data	MissForest	0.1328	0.0612	0.0493	0.0584	0.1998
	Matrix Factorization	0.1633	0.0816	0.0593	0.0737	0.1821
	GAIN	0.1583	0.0664	0.0471	0.0616	0.2248
	PC-GAIN	0.1327	0.0617	0.0472	0.0673	0.1937
	STGAN	0.1444	0.0679	0.0351	**0.0470**	0.1983
	BlockEcho(ours)	**0.1322**	**0.0608**	**0.0321**	0.0488	**0.1797**

Table 2: RMSE performance comparison of different data imputation methods at a fixed missing rate of 60%. Noted that in Movie dataset, 80% of the data are inherently missing, so we choose to mask the remaining visible elements at a missing rate of 60%.



(a) Block-wise missing data

(b) Scattered missing data

Figure 3: RMSE performance trend with increasing missing rate

fully designed experiments using various real-world datasets available in the public domain. The initial set of experiments compare BlockEcho with state-of-the-art baselines, focusing on a fixed missing rate as high as 60%. Following this, further experiments are conducted to assess the stability of these models as the missing rate is incrementally increased. Additionally, an ablation study is performed to examine the impact of specific components of the method. Finally, we designed a downstream prediction task using the imputed data as input, to evaluate how the performance of the data imputation method influences the end-to-end performance of a real-world prediction task. The experiments in this project are based on the following open-source real-world datasets from three distinct fields:

1. *Traffic Datasets (Time-Street matrix).* We employ two traffic flow datasets, ME-LA and PE-BAY, which collected in the highway of Los Angeles County and from California Transportation Agencies Performance Measurement System[Li *et al.*, 2017]. The traffic data are characterised by strong periodicity such as peak hours.
2. *COVID-19 Dataset (Date-City matrix).* This government dataset records daily COVID-19 cases (Cov-ca) and deaths (Cov-de) in major cities around the world since 2020. Infectious disease datasets are characterized by local continuity and sudden bursts.
3. *Movie Dataset (User-Movie matrix).* This dataset (Movie) records movie ratings collected from the MovieLens website[Harper and Konstan, 2015], which contains a large amount of missing data.

For "scattered missing data," we generate a random matrix of the same size as the original data and rank the random numbers within it. We compare the rankings with $missrate{*}m{*}n$ to determine which original elements should be masked. As for the "block-wise" missing data, we randomly select the top-left and bottom-right corners of a "block" within the feasible region limited by the missing rate, and then mask the original elements within that region.

All models were trained on an Nvidia Tesla V100S PCIE GPU and each experiment is repeated for ten times with different random seeds, and the results are averaged.

6.1 Performance of BlockEcho

In the first series of experiments, we synthetically mask 60% of the data in each dataset to generate a missing data matrix, and we compare the performance of BlockEcho with 5 baseline matrix completion models - MissForest[Stekhoven

	Method	ME-LA	PE-BAY	Cov-ca	Cov-de	Movie
Block-wise Missing Data	G+D_I+loss 3	0.1654	0.0793	0.0529	0.0798	–
	G+D_{II}+loss 3	0.1708	0.0825	0.0548	0.0879	–
	G+D_I+D_{II}	0.2651	0.2082	0.1945	0.2261	–
	G+D_I+D_{II}+MSE loss	0.1734	0.0818	0.0513	0.0797	–
	Ours(G+D_I+D_{II}+loss 3)	**0.1561**	**0.0743**	**0.0495**	**0.0784**	–
Scattered Missing Data	G+D_I+loss 3	0.1409	0.0631	0.0346	0.0510	0.1916
	G+D_{II}+loss 3	0.1419	0.0693	0.0408	0.0585	0.1921
	G+D_I+D_{II}	0.2071	0.1602	0.1322	0.1766	0.2810
	G+D_I+D_{II}+MSE loss	0.1398	0.0668	0.0363	0.0564	0.1893
	Ours(G+D_I+D_{II}+loss 3)	**0.1322**	**0.0608**	**0.0321**	**0.0488**	**0.1797**

Table 3: RMSE performance in the ablation study. The design of experiments is the same as 6.1

and Bühlmann, 2011], Matrix Factorization[Hastie *et al.*, 2015], GAIN[Yoon *et al.*, 2018], PC-GAIN [Wang *et al.*, 2021] and STGAN[Yuan *et al.*, 2022] — in terms of imputation accuracy. Data are masked in two ways: block-wise mask and scattered mask. To measure the imputation accuracy, we use RMSE to calculate the error between the model-imputed data and the original masked data, which is defined as $RMSE = \frac{||(\bar{X}-X)\odot(1-M)||_F}{\sum_{i,j}(1-M)_{ij}}$. In order to have a relatively uniform measure for different datasets, we normalize the data before computing RMSE. Table 2 reports the RMSE for BlockEcho and 5 other imputation models. Results show that BlockEcho performs significantly better than baselines, especially for block-wise missing data.

6.2 Models Performance in Different Missing Rates

In the second series of experiments, we take the traffic dataset PE-BAY as a representative, dynamically adjust the data missing rate from 20% to 80%, and explore the variation trend of each model performance with the data missing rate. Figure 3 quantitatively shows this trend: the SOTA machine learning algorithms represented by GAIN perform better on datasets with a low data missing rate, but will deteriorate rapidly as the missing rate increases; the Matrix Factorization method performs more stable but has a lower ceiling for accuracy. BlockEcho absorbs the advantages of both, and thus gives the best performance at high missing rates. Although at low missing rates it is not as accurate as some SOTA models, BlockEcho significantly outperforms them at high missing rates when the datasets become more incomplete. This superiority is more obvious in block-wise missing datasets.

6.3 Ablation Study

In this subsection we design ablation experiments to verify the contribution of each component of BlockEcho to the results. The table 3 shows that when we remove any part or replace it with a more conventional alternative, it will negatively affect the results. It is worth mentioning that when the matrix factorization loss function 3 is removed and only the GAN framework is used for matrix completion, the accuracy of the imputed data will be significantly reduced, which is why most generative models (including ours) use direct loss functions to guide the convergence of GAN.

	Block-wise	Scattered
Ori-Data	**0.0378**	**0.0378**
MissForest	0.0696	0.0512
Matrix Factorization	0.0615	0.0619
GAIN	0.0789	0.0484
PC-GAIN	0.0702	0.0479
STGAN	0.0685	0.0503
BlockEcho(ours)	**0.0484**	**0.0473**

Table 4: WMAPE of prediction task after various imputation model.

6.4 Case Study: Traffic Forecasting

In the final series of experiments, we design a downstream prediction task to illustrate how the performance of the data imputation method affects the end-to-end performance of a real-world prediction task. We take PE-BAY dataset as input to forecast the traffic conditions at the next timestamp. For the input datasets, we take the original data, the imputed data with our model BlockEcho, and the imputed data from each baseline model for comparison. For traffic forecasting, we use Random Forests (RF) with the same set of hyperparameter settings with the same feature engineering for all input datasets. We use Weighted Mean Absolute Percentage Error (WMAPE) as the error metric to measure the prediction performance. Table 4 summarizes the prediction results. Comparing Table 2 and Table 4, we find that models with higher data imputation accuracy tend to give better results in subsequent prediction tasks, and BlockEcho, the best-performing data imputation model, also gives the lowest prediction errors in subsequent traffic forecasting tasks.

7 Conclusion and Future Work

In this paper, we excavate and mathematically define the issue of "block-wise" missing data and innovatively propose the solution, BlockEcho. Experiments on various data sets, especially with block missing data, show that our method outperforms other SOTA methods. Our future work will extend to federated learning where block-wise missing data widely appear.

Acknowledgments

This work was supported by National Key R&D Program of China (2022YFB4501500,2022YFB4501504).

References

[Buuren and Oudshoorn, 2000] S. Van Buuren and C. G. M Oudshoorn. Multivariate imputation by chained equations : Mice v1.0 user's manual. 2000.

[Chang *et al.*, 2023] Zhuoqing Chang, Shubo Liu, Zhaohui Cai, and Guoqing Tu. Anode-gan: Incomplete time series imputation by augmented neural ode-based generative adversarial networks. In *International Conference on Artificial Neural Networks*, pages 16–27. Springer, 2023.

[Gao *et al.*, 2020] Xin Gao, Fang Deng, and Xianghu Yue. Data augmentation in fault diagnosis based on the wasserstein generative adversarial network with gradient penalty. *Neurocomputing*, 396:487–494, 2020.

[Gold and Bentler, 2000] Michael Steven Gold and Peter M Bentler. Treatments of missing data: A monte carlo comparison of rbhdi, iterative stochastic regression imputation, and expectation-maximization. *Structural Equation Modeling*, 7(3):319–355, 2000.

[Goodfellow *et al.*, 2020] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[Harper and Konstan, 2015] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.

[Hastie *et al.*, 2015] Trevor Hastie, Rahul Mazumder, Jason D Lee, and Reza Zadeh. Matrix completion and low-rank svd via fast alternating least squares. *The Journal of Machine Learning Research*, 16(1):3367–3402, 2015.

[Kong *et al.*, 2023] Xiangjie Kong, Wenfeng Zhou, Guojiang Shen, Wenyi Zhang, Nali Liu, and Yao Yang. Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data. *Knowledge-Based Systems*, 261:110188, 2023.

[Lee and Seung, 2000] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[Lee *et al.*, 2019] Dongwook Lee, Junyoung Kim, Won-Jin Moon, and Jong Chul Ye. Collagan: Collaborative gan for missing image data imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2487–2496, 2019.

[Li *et al.*, 2017] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[Li *et al.*, 2019] Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. Misgan: Learning from incomplete data with generative adversarial networks. *arXiv preprint arXiv:1902.09599*, 2019.

[Mescheder *et al.*, 2017] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *Advances in neural information processing systems*, 30, 2017.

[Miyato *et al.*, 2018] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[Neves *et al.*, 2021] Diogo Telmo Neves, Marcel Ganesh Naik, and Alberto Proença. Sgain, wsgain-cp and wsgain-gp: Novel gan methods for missing data imputation. In *International Conference on Computational Science*, pages 98–113. Springer, 2021.

[Santos *et al.*, 2019] Miriam Seoane Santos, Ricardo Cardoso Pereira, Adriana Fonseca Costa, Jastin Pompeu Soares, João Santos, and Pedro Henriques Abreu. Generating synthetic missing data: A review by missing mechanism. *IEEE Access*, 7:11651–11667, 2019.

[Scheffer, 2002] Judi Scheffer. Dealing with missing data. 2002.

[Stekhoven and Bühlmann, 2011] D. J. Stekhoven and P Bühlmann. Missforest - nonparametric missing value imputation for mixed-type data. 2011.

[Wang *et al.*, 2021] Yufeng Wang, Dan Li, Xiang Li, and Min Yang. Pc-gain: Pseudo-label conditional generative adversarial imputation networks for incomplete data. *Neural Networks*, 141:395–403, 2021.

[Williams, 2015] R Williams. Missing data part 1: Overview, traditional methods. university of notre dame, 2015.

[Yoon and Sull, 2020] Seongwook Yoon and Sanghoon Sull. Gamin: Generative adversarial multiple imputation network for highly missing data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8456–8464, 2020.

[Yoon *et al.*, 2018] Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pages 5689–5698. PMLR, 2018.

[Yu *et al.*, 2016] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems*, 29, 2016.

[Yuan *et al.*, 2022] Ye Yuan, Yong Zhang, Boyue Wang, Yuan Peng, Yongli Hu, and Baocai Yin. Stgan: Spatio-temporal generative adversarial network for traffic data imputation. *IEEE Transactions on Big Data*, 2022.

[Zhang *et al.*, 2021] Weibin Zhang, Pulin Zhang, Yinghao Yu, Xiying Li, Salvatore Antonio Biancardo, and Junyi Zhang. Missing data repairs for traffic flow

with self-attention generative adversarial imputation net. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):7919–7930, 2021.