Masked Autoencoder for Data Recovery in Polymer Research: Mitigating Data Integrity Threats

Bruno Saidou Fonkeng*, Fuhao Li*, Binglin Sui†, and Jielun Zhang*
* School of Electrical Engineering & Computer Science, University of North Dakota, Grand Forks, ND, 58202, USA
† Department of Chemistry, University of North Dakota, Grand Forks, ND, 58202, USA
Emails: {*bruno.fonkeng, *fuhao.li, †binglin.sui, *jielun.zhang}@UND.edu

2024 Cyber Awareness and Research Symposium (CARS) | 979-8-3503-8641-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/CARS61786.2024.10778772

***Abstract*—Polymer research critically depends on high-quality datasets to understand material properties, develop new materials, and optimize existing ones. However, these datasets are increasingly vulnerable to data integrity threats such as accidental deletions, system failures, and cyber attacks, leading to significant data loss and corruption. Traditional data recovery methods, including mean and median imputation, often fail to accurately restore missing data due to their inability to capture complex data relationships. While machine learning techniques like autoencoders offer improved recovery by learning intricate data patterns, their effectiveness in polymer research data remains largely unvalidated. In this paper, we propose an autoencoder-based data recovery approach that utilizes random masks to guide the recovery process. This method is specifically tailored to the complexities of polymer datasets, aiming to achieve higher accuracy and robustness in data recovery. We evaluated the proposed approach using cloud point temperature datasets for binary polymer solutions. The evaluation results demonstrate it outperforms conventional imputation methods, effectively mitigating the consequence of data loss and compromised data integrity.**

I. Introduction

Polymer research relies heavily on comprehensive datasets that provide detailed information about various polymer properties, including molecular weight distribution, thermal stability, mechanical strength, and chemical resistance [1]. These datasets are critical for scientists to understand the behavior of polymers under different conditions, develop new materials with tailored properties, and optimize existing polymers for various industrial applications [2]. During the synthesis and testing phases, data is meticulously collected to capture a broad spectrum of parameters that influence polymer characteristics. High-quality datasets allow researchers to build predictive models, facilitate the discovery of novel materials, and improve the overall efficiency of polymer development [3].

However, the integrity and security of polymer datasets are increasingly threatened by multiple factors, posing significant risks to the reliability and validity of research outcomes. Data integrity threats can arise from a range of incidents, including accidental data deletion, hardware or software failures, and, more alarmingly, targeted cyber-attacks [4]. These threats can lead to substantial data loss or corruption, resulting in incomplete or misleading datasets. The consequences of such integrity breaches are severe, potentially causing delays in research, increased costs due to the need for repeated experiments, and the dissemination of incorrect scientific conclusions, which could misguide future research and applications. Moreover, unauthorized modifications or data breaches can compromise proprietary information, leading to competitive disadvantages and potential intellectual property theft [5].

To mitigate these risks, various data recovery methods have been developed over the years. Conventional approaches, such as mean imputation, median imputation, are commonly employed to handle missing data by substituting missing values with statistical averages or the most frequent occurrences. Although these methods are easy to implement, they often fail to account for the underlying feature relationships, consequently leading to biased or inaccurate imputations. More advanced techniques, including Machine Learning and Deep Learning based approaches such as autoencoders [6], Recurrent Neural Network (RNN) [7], and diffusion models [8], have been proposed to overcome these limitations. For example, [7] introduced a Dynamic Layered-Recurrent Neural Network approach for recovering missing data in the Internet of Medical Things applications, offering effectiveness and efficiency. However, these techniques still face challenges, for example, sometimes failing to generalize well to new, unseen datasets. Additionally, the effectiveness of the aforementioned methods has not yet been thoroughly validated within the specific context of polymer research data. Polymer datasets often exhibit unique patterns and complexities that these methods may not adequately address, leading to potential inaccuracies in data recovery and limitations in their applicability to this specialized domain.

In this paper, we propose a novel autoencoder-based data recovery approach specifically designed to address the challenges of missing data in polymer research datasets [9]. Our method incorporates manually crafted masks that guide the autoencoder in identifying and recovering missing data points, enhancing its ability to handle datasets with varying levels of data loss. As a proof of concept, our approach is applied in recovering missing data related to cloud point temperature studies of binary polymer solutions [10]. The performance of our method is evaluated against conventional imputation techniques and other advanced approaches using multiple publicly available datasets. The results demonstrate that our proposed approach significantly outperforms existing ones in terms of data recovery accuracy and robustness, significantly minimizing the impact of data loss and enhancing the integrity of the recovered datasets.

The rest of this paper is organized as follows. Section II describes the related work on data recovery methods. Section III formulates the problem of missing data in polymer research. Section IV presents the proposed autoencoder-based data recovery approach, including the design of the masking process and the training process. Section V presents the experiment settings and the experiment results. Section VI concludes the paper with a summary of our findings and future work.

II. Related Work

A. Traditional and Statistical Imputation Methods

Data recovery and imputation have long been studied across various scientific fields, where missing data can critically affect research outcomes. Traditional methods, such as mean, median, and mode imputation, are popular due to their simplicity and ease of implementation [11]–[13]. These approaches replace missing values with statistical estimates from the observed data, assuming that such replacements will not significantly alter the data distribution. However, this assumption often does not hold, especially in fields like polymer research, where data relationships are highly nonlinear and feature-dependent. For instance, mean imputation, which assumes randomness in missing values, can lead to biased results and fail to capture complex dependencies within the data.

Statistical techniques, such as Multiple Imputation by Chained Equations (MICE) and Expectation-Maximization (EM), have also been developed to improve upon these simplistic approaches [14], [15]. These methods enhance imputation accuracy but can be limited by the complexity and high computational costs. However, they still face significant limitations, for example, when applied to large-scale datasets with complex interdependencies.

B. Artifical Intelligence based Approaches

Recent advancements in Machine Learning algorithms and Deep Learning models have been offering new ways to handle complex, high-dimensional data [6], [8]. Models such as autoencoders, Recurrent Neural Networks (RNNs), and diffusion models have demonstrated notable improvements in data recovery. Specifically, autoencoders are particularly effective at learning compressed representations of data and reconstructing it, which helps in capturing and restoring intricate patterns and feature relationships. RNNs are capable of managing sequential or time-series data by leveraging their ability to model temporal dependencies and sequential structures. Diffusion models have emerged as powerful tools for data generation and imputation by modeling complex data distributions through advanced generative processes. These methods enhance the accuracy and efficiency of data recovery, offering significant advantages over traditional techniques. For example, the authors in [6] proposed a siamese autoencoder-based approach for imputation to address the issue of missing data imputation. This method incorporates a deep autoencoder architecture, a custom loss function, and a triplet mining strategy, and it outperformed existing methods across 14 heterogeneous healthcare datasets. The authors in [8] introduced a diffusion model based algorithm for tabular data imputation, which uses a conditioning attention mechanism and an encoder-decoder transformer as the denoising network. Their model showed improvements in learning efficiency, statistical similarity, and privacy risk mitigation, particularly excelling in handling missing data imputation and synthetic data generation. However, these adaptations have not been thoroughly validated in the context of polymer research.

III. Problem Statement

Let $\mathcal{D} = \{x_{ij}\}$ denote a dataset where each entry x_{ij} represents a specific measurement or property of the polymer. Due to incidents such as accidental deletions, system failures, or malicious cyber-attacks, certain entries in $\mathcal{D}$ may be lost or corrupted. With such an assumption, we denote $\mathcal{M} = \{m_{ij}\}$ to be a binary mask matrix of the same dimension as $\mathcal{D}$, where:

$$m_{ij} = \begin{cases} 1 & \text{if } x_{ij} \text{ is observed,} \\ 0 & \text{if } x_{ij} \text{ is missing or corrupted.} \end{cases}$$

The goal of this research is to recover the missing or corrupted data entries $\{x_{ij} : m_{ij} = 0\}$ in the dataset $\mathcal{D}$. Formally, we aim to find an imputed dataset $\hat{\mathcal{D}}$ that minimizes the reconstruction error between the observed data and the imputed data. This can be expressed as the following optimization problem:

$$\min_{\hat{\mathcal{D}}} \mathcal{L}(\mathcal{D}, \hat{\mathcal{D}}, \mathcal{M}) = \sum_{i,j} m_{ij} \ell(x_{ij}, \hat{x_{ij}}), \tag{1}$$

where ℓ represents the loss function, e.g., mean squared error, that quantifies the difference between the observed data x_{ij} and the imputed data $\hat{x_{ij}}$ for all (i, j) such that $m_{ij} = 1$.

IV. Methodology

In this section, we describe the proposed autoencoder-based data recovery approach based on a remasking technique [9]. The proposed approach is designed to address the challenges posed by incomplete or corrupted data entries, ensuring high accuracy in data recovery while preserving the integrity and structure of the original polymer research dataset. Specifically, an autoencoder architecture combined with a masking strategy to is employed to improve the reconstruction of missing values in tabular datasets. The core idea is to train the autoencoder not only to reconstruct naturally missing values but also to generalize across different patterns of missingness by dynamically masking observed data during the training phase. This strategy forces the model to learn a more generalized and robust representation of the underlying data distribution.

A. Transformer based Encoder and Decoder

In specific, the autoencoder in the proposed approach consists of an encoder and a decoder, where the encoder, denoted as f_θ, encodes the input data into latent representations, and the decoder, denoted as d_ϕ, reconstructs the input data from the latent space. On top of the standard autoencoder setting up, the approach applies a dynamic re-masking strategy during

training phase to enhance the recovery robustness. The detailed descriptions of these components are given as follows.

Let $\mathbf{x}_i = [x_1, x_2, \ldots, x_L]$ to be the feature vector representing a row of data instance that contains values of L features, the encoder processes the input feature set $\mathbf{x}_i$ to generate a latent representation $\mathbf{z}_i$, expressed as $\mathbf{z}_i = f_\theta(\mathbf{x}_i)$. This process involves passing $\mathbf{x}_i$ through multiple Transformer blocks, where each block applies the multi-head self-attention mechanism along with a subsequent feed-forward network. The self-attention mechanism enables the model to focus on various segments of the input sequence to capture the relationships between features [16]. The output of each Transformer block is then fed into the next block, gradually refining the latent representation $\mathbf{z}_i$. The decoder takes this latent representation $\mathbf{z}_i$ and also employs a sequence of Transformer blocks to construct $\hat{\mathbf{x}}_i$ to mimic the original feature vector $\mathbf{x}_i$, given by $\hat{\mathbf{x}}_i = d_\phi(\mathbf{z}_i)$.

B. Dynamic Re-Masking in Batch Learning

To prevent overfitting and improve generalization during the batch learning phrase, a re-mask matrix $\mathcal{M}' \in \{0,1\}^{N\times L}$, where N is number of input data in a batch, will be generated randomly. Note that we initially have a mask matrix $\mathcal{M}$ that indicates the original masking state of the dataset, i.e., $m_{ij} = 1$ indicates the observed value, and $m_{ij} = 0$ indicates the missing value. During batch learning, for each iteration, there will be a subset of the observed data extracted from the data batch, and it will be masked out using $\mathcal{M}'$:

$$m'_{ij} = \begin{cases} 1 & \text{if the value is to remain unmasked,} \\ 0 & \text{if the value is selected to be masked.} \end{cases}$$

In this case, this re-masking process creates two subsets for each sample, i.e., location matrix of re-masked values $\mathcal{I}_{\text{remask}} = \{i, j : m_{ij} = 1 \wedge m'_{ij} = 0\}$ and location matrix of unmasked values $\mathcal{I}_{\text{unmask}} = \{i, j : m_{ij} = 1 \wedge m'_{ij} = 1\}$. Ultimately, the autoencoder aims to minimize the reconstruction error for the re-masked values, and a regularization term is added to promote the performance of the overall reconstruction considering both the re-masked values and the unmasked values. The overall loss funtion is defined as follows.

$$\mathcal{L}(\theta,\phi) = \frac{1}{|\mathcal{I}_{\text{remask}}|}\sum_{i=1}^{n}\sum_{j\in\mathcal{I}_{\text{remask}}}(x_{ij}-\hat{x}_{ij})^2 + \lambda\frac{1}{|\mathcal{I}_{\text{unmask}}|}\sum_{i=1}^{n}\sum_{j\in\mathcal{I}_{\text{unmask}}}(x_{ij}-\hat{x}_{ij})^2. \quad (2)$$

where λ is a coefficient that controls the impact of the regularization term.

C. Imputation of Missing Values

The remasking technique is omitted in the imputation stage. The encoder processes only the observed data to generate latent representations$\mathbf{z}_i$. The decoder then reconstructs the missing or corrupted entries in the original dataset $\mathcal{D}$ directly from these representations. The reconstructed values are subsequently used to fulfill the missing entries, integrating with the observed values, to complete the original dataset $\mathcal{D}$. The detailed procedures in the proposed batch learning and the imputation stages are summarized in Alg. 1.

Algorithm 1 Autoencoder-based missing data imputation with remasking strategy.

Require: Dataset $\mathcal{D}$ with missing values, hyperparameter λ.
Ensure: Imputed dataset $\hat{\mathcal{D}}$.

Initialization: Initialize θ in encoder $f_\theta(\cdot)$, ϕ in decoder $d_\phi(\cdot)$, and compute $\mathcal{M}$ in $\mathcal{D}$.
procedure FITTING PHASE
  for each data batch **do**
    Generate remask matrix $\mathcal{M}'$.
    Compute location matrix: $\mathcal{I}_{\text{remask}}, \mathcal{I}_{\text{unmask}}$.
    Process unmask input batch with $\mathcal{I}_{\text{unmask}}$ by encoder $f_\theta(\cdot)$ and obtain latent representation.
    Reconstruct the input batch using decoder $d_\phi(\cdot)$.
    Compute reconstruction loss $\mathcal{L}(\theta,\phi)$ using Eq. (2).
    Update encoder and decoder parameters θ, ϕ using gradient descent algorithm.
  end for
end procedure
procedure IMPUTATION PHASE
  for each sample $\mathbf{x}_i \in \mathcal{D}$ **do**
    Compute latent representation using $f_\theta(\cdot)$.
    Reconstruct the values using decoder $d_\phi(\cdot)$.
  end for
  Compose the imputed dataset $\hat{\mathcal{D}}$ using the reconstructed values and the observed values in $\mathcal{D}$.
end procedure

V. EXPERIMENT

A. Datasets

To evaluate our proposed scheme, we utilized a subset of the liquid-liquid equilibrium data for binary polymer solutions from the CRC Handbook of Liquid-Liquid Equilibrium Data of Polymer Solutions [17], which was further preprocessed as described in [10], [18]. This dataset includes 6,524 cloud point data points, covering 21 polymers, 61 solvents, and 97 unique polymer-solvent systems. We selected several datasets that have the largest sizes among all the others to evaluate our algorithm to avoid loss of generality. We partitioned the dataset based on ref in the dataset which is shown in Table I.

In specific, we perform data pre-processing by extracting and reorganizing the original features into eight features, namely polymer CAS registry number , solvent CAS registry number, polymer weight-average molecular weight in grams per mole, polydispersity index, polymer volume fraction, polymer mass fraction, pressure in megapascals, and cloud point temperature in degrees Celsius. It is worth mentioning that we have only used complete data instances in the datasets as we need those original values as the ground truth for performance evaluation, and the data were then encoded and normalized via the Min-max approach [19].

TABLE I
DESCRIPTION OF SELECTED DATASETS.

Notation	Size	References
Dataset 1	531	doi.org/10.1006/jcht.1999.0607
Dataset 2	497	doi.org/10.1021/ma9517308
Dataset 3	317	doi.org/10.1021/ma00107a011
Dataset 4	264	doi.org/10.1016/S0378-3812(97)00157-X
Dataset 5	252	doi.org/10.1016/j.supflu.2005.08.004
Dataset 6	199	doi.org/10.1063/1.464440
Dataset 7	184	doi.org/10.1002/(SICI)1099-0488 (199703)35:4<631::AID-POLB11>3.0.CO;2-H
Dataset 8	179	doi.org/10.1016/0032-3861(95)91454-F
Dataset 9	139	doi.org/10.1021/ma00015a024
Dataset 10	138	1986KRU in ISBN: 9781420067989

B. Experiment Settings

Our method adopts a Transformer based autoencoder architecture design, where the encoder includes an embedding layer that maps the input data into a 16-dimensional embedding space, with positional embeddings added to retain sequence information, 4 Transformer blocks, each containing 4 attention heads. The decoder uses a linear layer to transform the output of the encoder to the decoder's input embedding dimension, followed by 4 Transformer blocks to perform the decoding, and another linear layer to restore the output to the original data shape. The model employs a random masking strategy, where parts of the input data are randomly masked during training. The λ in Eq. (2) has been set to 1 to balance the impact of the loss computed on both masked and unmasked elements. Other hyperparameters are set as follows, where batch size is 256, embedding dimension is 16, training epoch number is 1000, the initial learning rate is 0.002, and it drops 0.5% for every 50 epochs.

We simulated missing data by setting certain values to NaN. Specifically, given a NaN ratio, the dataset was randomly altered by setting features to NaN according to this ratio. After training, the model's performance was evaluated by comparing the predicted values for these NaN entries with their original values. The simulations and evaluations were carried out on a workstation featuring an Intel® Core™ i7-14700K processor, 32 GB of RAM, a 4 TB SSD, and an NVIDIA GeForce RTX 4070. The simulation and the development of the proposed schemes were implemented using PyTorch 2.3.0 and Python 3.11.9.

C. Compared Imputation Methods

To benchmark the performance of our proposed autoencoder-based data recovery approach, we compared it against several traditional and advanced imputation methods. These methods are commonly used in the literature and serve as a baseline for evaluating the effectiveness of new data recovery techniques. The imputation methods considered in this study are as follows:

1) Mean Imputation [11] : This method replaces missing values with the mean of the observed data for a particular feature.

2) Median Imputation [12]: This approach substitutes missing values with the median of the observed data.

3) Frequent Imputation [13]: Also known as mode imputation, this method fills in missing data with the most frequently occurring value in a feature.

4) SoftImpute Imputation: SoftImpute uses matrix factorization to impute missing values. It models the data matrix as a low-rank approximation, iteratively filling in the missing entries based on observed data, making it effective for structured data scenarios.

5) MICE Imputation [14]: MICE iteratively estimates missing values by generating multiple possible imputed datasets. Each iteration uses the previous results as a starting point, providing a flexible and diverse approach to imputation.

6) EM Imputation [15]: EM algorithm iteratively estimates missing data and model parameters by maximizing the likelihood function. It first computes the expected values for the missing data and then updates the model parameters based on these estimates until convergence.

D. Metrics

To evaluate the performance of our data recovery approach, we employed two key metrics: Root Mean Squared Error and Wasserstein Distance. These metrics are widely used in data imputation and recovery studies to assess the accuracy and robustness of recovery methods.

1) Root Mean Squared Error: The Root Mean Square Error (RMSE) measures the square root of the average squared differences between original and recovered data points, providing an intuitive metric for recovery accuracy. Lower RMSE values indicate smaller discrepancies and better performance. It is calculated as RMSE $= \left[\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right]^{1/2}$, where n is the number of data points, $\hat{y}_i$ is the recovered value, and y_i is the true value.

2) Wasserstein Distance: The Wasserstein Distance (WD), also known as Earth Mover's Distance, quantifies the dissimilarity between two probability distributions, assessing the alignment between original and recovered data distributions [20]. Lower WD values indicate better alignment. It is expressed as $W(p,q) = \inf_{\gamma\in\Pi(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$, where p and q are the distributions of original and recovered data, and $||x-y||$ represents the Euclidean distance between points x and y.

E. Evaluation Results

Table II presents the performance of RMSE and WD across 10 selected datasets under different masking ratios, i.e., 0.15, 0.3, 0.45 and 0.6. Generally speaking, both RMSE and WD tend to increase as the masking ratio increases within the several datasets. We can see that the RMSE and WD with a making ratio of 0.15 are the lowest in most datasets, i.e., Dataset 2, Dataset 3, Dataset 5, Dataset 6, Dataset 9, and Dataset 10. For instance, in Dataset 2, when the masking ratio is 0.15, the RMSE and WD are 0.126 and 0.031, respectively, and sightly rise to 0.127 and 0.034. When the masking ratio is 0.3. When the masking ratio is 0.45, the RMSE and WD

TABLE II
AVERAGE RMSE AND WD FOR DIFFERENT MASKING RATIOS ACROSS DATASETS

Masking ratio	Dataset 1		Dataset 2		Dataset 3		Dataset 4		Dataset 5		Dataset 6		Dataset 7		Dataset 8		Dataset 9		Dataset 10	
	RMSE	**WD**	**RMSE**	**WD**	**RMSE**	**WD**	**RMSE**	**WD**	**RMSE**	**WD**	**RMSE**	**WD**	**RMSE**	**WD**	**RMSE**	**WD**	**RMSE**	**WD**	**RMSE**	**WD**
0.15	0.128	0.04	**0.126**	**0.031**	**0.115**	**0.032**	0.12	0.033	**0.071**	**0.014**	**0.104**	**0.026**	0.144	0.036	0.203	0.049	**0.149**	**0.04**	**0.047**	**0.015**
0.3	**0.115**	0.039	0.127	0.034	0.124	0.032	0.093	**0.022**	0.077	0.021	0.11	0.033	**0.135**	**0.034**	**0.18**	**0.042**	0.149	0.041	0.083	0.029
0.45	0.121	**0.037**	0.143	0.033	0.123	0.034	**0.088**	0.027	0.083	0.02	0.122	0.029	0.149	0.04	0.226	0.055	0.162	0.054	0.099	0.028
0.6	0.168	0.05	0.168	0.041	0.156	0.04	0.122	0.034	0.089	0.035	0.152	0.041	0.143	0.04	0.249	0.082	0.221	0.071	0.082	0.021

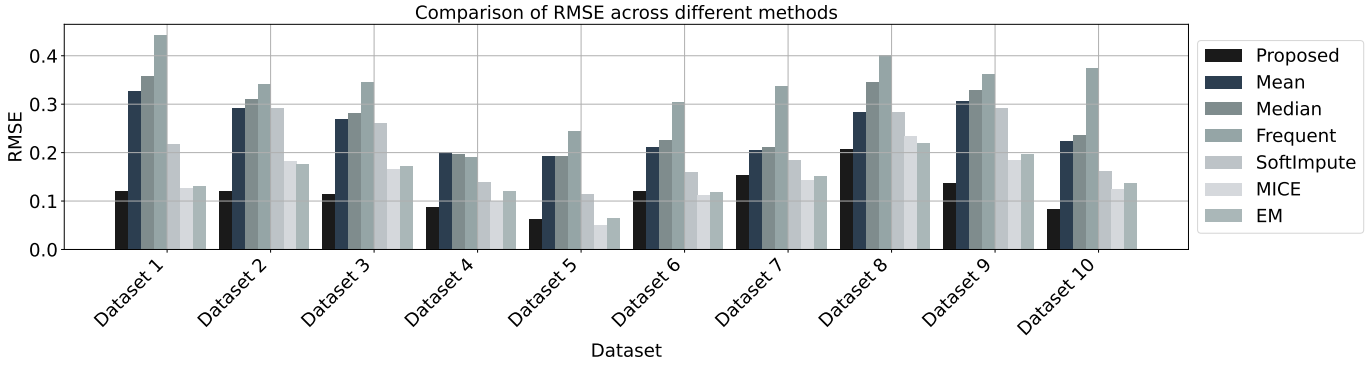

Fig. 1. Bar chart comparing RMSE values of various data imputation methods across ten different datasets (Dataset 1 to Dataset 10). The x-axis lists the datasets, while the y-axis shows the RMSE values, which indicate the imputation error for each method. Seven different methods are compared: Proposed, Mean, Median, Frequent, SoftImpute, MICE, and EM, represented by different shades of gray.

increase to 0.143 and 0.033, respectively, and reach the highest value of 0.168 and 0.041 when the masking ratio is 0.6. Similarly, in Dataset 9, when the masking ratio is set to 0.15, the RMSE and WD are 0.149 and 0.04, respectively,,and keeps increasing with an increasing making ratio, The RMSE and WD are 0.221 and 0.071 which are the highest when the masking ratio is 0.6.

Fig. 1 illustrates the RMSE performance of our proposed method compared to existing methods across 10 datasets. It is worth mentioning that we decided to choose a masking ratio of 0.15 Based on the overall performance of the masking ratio. Besides our proposed method, Mean, Median, Frequent, SoftImpute, MICE, and EM are developed for comparison. The missing ratio is set to 0.1. As shown in the figure, the proposed method generally achieves the lowest RMSE across most datasets. For instance, in Dataset 3, the proposed method outperforms the other methods with RMSE values of 0.115, MICE and EM show the higher RMSE which are 0.166 and 0.173. The Mean, Median, and SoftImpute have similar RMSEs which are 0.269, 0.282, and 0.261, respectively, and Frequent shows the highest RMSE which is 0.346. In contrast, methods like Frequent-based imputation and SoftImpute algorithm tend to lead higher RMSE values in several datasets, particularly in Dataset 1, Dataset 2, and Dataset 8, where the RMSE values are around 0.4.

Fig. 2 shows the box plots showing the RMSE performance of our proposed method under different missing ratios, with the fixed masking ratio of 0.15. In this figure, we can see that the RMSE values for most datasets show an upward trend when the missing ratio increases. For example, In Dataset 3, the RMSE gradually increases from 0.11 at a missing ratio of 0.1 to 0.18, 0.215, and 0.26 at a missing ratio of 0.2, 0.3, and 0.4, respectively. In addition, the variability shown in the figure are vary when using different datasets. For example, Datasets 2 shows relatively small variability in the RMSE box plots, especially when the missing ratio is set to 0.1. The corresponding Minimum, lower quartile, upper quartile, and Maximum are 0.105, 0.11, 0,125 and 0.13, respectively. When the missing ratio is 0.4, they rise to 0.22, 0.25, 0.28, and 0.3, which shows a higher variability. Overall, low missing Ratio means that there are more known data points, which allows for obtaining a lower and more stable RMSE. while a higher, missing ratio leads to increased RMSE and instability due to the gradual loss of data, indicating an increase in distortion during the data recovery process and reflecting the impact of data integrity loss on recovery quality.

VI. CONCLUSION

In conclusion, this study presents a novel approach to data recovery tailored specifically for the complexities of polymer research datasets. The proposed method utilizes an autoencoder-based framework enhanced by manually crafted masks, which guide the recovery process to achieve higher accuracy and robustness. Unlike traditional imputation methods such as mean and median replacement, which fail to capture the intricate relationships within polymer data, our approach leverages the power of machine learning to model these complex patterns effectively. The application of this technique to cloud point temperature datasets for polymer-solvent solutions has shown that it significantly outperforms

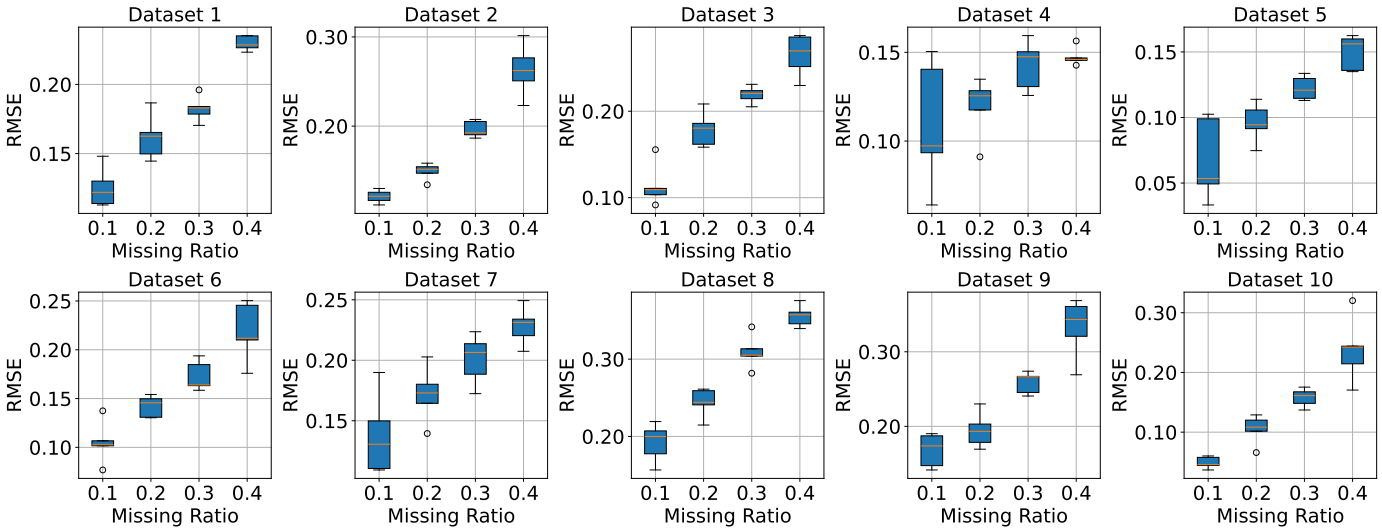

Fig. 2. Box plots showing the RMSE across different datasets (Dataset 1 to Dataset 10) for varying missing data ratios. The x-axis represents the Missing Ratio, which indicates the proportion of data missing in each dataset, ranging from 0.1 to 0.4. The circular data points in each box plot represent outliers that are observations falling outside the whisker range.

both conventional and existing AI-based recovery methods. By effectively minimizing data loss and preserving the integrity of the recovered datasets, our method ensures that critical information necessary for understanding material properties, developing new materials, and optimizing existing ones is retained. This is particularly vital in polymer research, where data quality directly impacts the reliability of research outcomes and the development of new technologies. The success of our approach underscores the potential of machine learning techniques in advancing data recovery practices in scientific research, providing a foundation for more accurate and reliable dataset preservation. Future work will focus on extending the method to accommodate a broader range of polymer research datasets and integrating it with dynamic, real-time data recovery systems.

Acknowledgment

The authors also gratefully acknowledge the support from the Early Career Scholars award, provided by the Office of the Vice President for Research and Economic Development, University of North Dakota.

References

[1] D. J. Audus and J. J. de Pablo, "Polymer informatics: opportunities and challenges," *ACS macro letters*, vol. 6, no. 10, pp. 1078–1082, 2017.

[2] R. Ma and T. Luo, "Pi1m: a benchmark database for polymer informatics," *Journal of Chemical Information and Modeling*, vol. 60, no. 10, pp. 4684–4690, 2020.

[3] T. D. Huan, A. Mannodi-Kanakkithodi, C. Kim, V. Sharma, G. Pilania, and R. Ramprasad, "A polymer dataset for accelerated property prediction and design," *Scientific data*, vol. 3, no. 1, pp. 1–10, 2016.

[4] W. Liang, Y. Fan, K.-C. Li, D. Zhang, and J.-L. Gaudiot, "Secure data storage and recovery in industrial blockchain network environments," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6543–6552, 2020.

[5] N. Sharma, E. A. Oriaku, N. Oriaku *et al.*, "Cost and effects of data breaches, precautions, and disclosure laws," *International Journal of Emerging Trends in Social Sciences*, vol. 8, no. 1, pp. 33–41, 2020.

[6] C. Fu, M. Quintana, Z. Nagy, and C. Miller, "Filling time-series gaps using image techniques: Multidimensional context autoencoder approach for building energy data imputation," *Applied Thermal Engineering*, vol. 236, p. 121545, 2024.

[7] H. Turabieh, A. A. Salem, and N. Abu-El-Rub, "Dynamic l-rnn recovery of missing data in iomt applications," *Future Generation Computer Systems*, vol. 89, pp. 575–583, 2018.

[8] M. Villaizán-Vallelado, M. Salvatori, C. Segura, and I. Arapakis, "Diffusion models for tabular data imputation and synthetic data generation," *arXiv preprint arXiv:2407.02549*, 2024.

[9] T. Du, L. Melis, and T. Wang, "Remasker: Imputing tabular data with masked autoencoding," *arXiv preprint arXiv:2309.13793*, 2023.

[10] J. G. Ethier, R. K. Casukhela, J. J. Latimer, M. D. Jacobsen, A. B. Shantz, and R. A. Vaia, "Deep learning of binary solution phase behavior of polystyrene," *ACS Macro Letters*, vol. 10, no. 6, pp. 749–754, 2021.

[11] Z. Zhang, "Missing data imputation: focusing on single imputation," *Annals of translational medicine*, vol. 4, no. 1, 2016.

[12] G. F. Berkelmans, S. H. Read, S. Gudbjörnsdottir, S. H. Wild, S. Franzen, Y. Van Der Graaf, B. Eliasson, F. L. Visseren, N. P. Paynter, and J. A. Dorresteijn, "Population median imputation was noninferior to complex approaches for imputing missing values in cardiovascular prediction models in clinical practice," *Journal of Clinical Epidemiology*, vol. 145, pp. 70–80, 2022.

[13] E. Kreiner-Møller, C. Medina-Gomez, A. G. Uitterlinden, F. Rivadeneira, and K. Estrada, "Improving accuracy of rare variant imputation with a two-step imputation approach," *European Journal of Human Genetics*, vol. 23, no. 3, pp. 395–400, 2015.

[14] P. Royston, "Multiple imputation of missing values: update of ice," *The Stata Journal*, vol. 5, no. 4, pp. 527–536, 2005.

[15] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Computing and Applications*, vol. 19, pp. 263–282, 2010.

[16] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[17] C. Wohlfarth, *CRC handbook of liquid-liquid equilibrium data of polymer solutions*. CRC Press, 2007.

[18] J. G. Ethier, R. K. Casukhela, J. J. Latimer, M. D. Jacobsen, B. Rasin, M. K. Gupta, L. A. Baldwin, and R. A. Vaia, "Predicting phase behavior of linear polymers in solution using machine learning," *Macromolecules*, vol. 55, no. 7, pp. 2691–2702, 2022.

[19] S. Patro, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.

[20] V. M. Panaretos and Y. Zemel, "Statistical aspects of wasserstein distances," *Annual review of statistics and its application*, vol. 6, no. 1, pp. 405–431, 2019.