Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.082996

ARTICLE

EGAIN: Enhanced Generative Adversarial Networks for Imputing Missing Values

Abolfazl Saghafi[1,*], **Soodeh Moallemian**[2], **Miray Budak**[2] and **Rutvik Deshpande**[2]

[1]Department of Mathematics, Saint Joseph's University, Philadelphia, PA, USA
[2]Center for Molecular & Behavioral Neuroscience, Rutgers University–Newark, Newark, NJ, USA

*Corresponding Author: Abolfazl Saghafi. Email: asaghafi@sju.edu

Received: 26 March 2026; Accepted: 13 May 2026; Published: 15 June 2026

ABSTRACT: Missing data remain a persistent challenge in statistical analysis and machine learning because many predictive methods require complete observations. Generative Adversarial Imputation Networks (GAIN) offer a flexible deep-learning approach for missing value imputation, but their practical use is limited by convergence instability, sensitivity to hyperparameter selection, and dependence on outdated software implementations. To address these limitations, we propose Enhanced Generative Adversarial Imputation Networks (EGAIN), a modernized extension of GAIN implemented in TensorFlow 2.x. EGAIN incorporates convolution-based generator and discriminator networks, a channel-stacked representation of the data and mask, and checkpoint-based training diagnostics to improve stability and usability. EGAIN was evaluated on five benchmark datasets under multiple Missing Completely At Random (MCAR) settings and compared with the original GAIN implementation and median imputation. Across most evaluated conditions, EGAIN achieved lower root mean squared error (RMSE) and showed greater robustness, particularly when missingness was concentrated in a subset of variables. These results indicate that EGAIN provides a more stable and reproducible framework for missing data imputation in tabular datasets.

KEYWORDS: Missing value imputation; generative adversarial network; tabular data imputation; missing completely at random; convolutional architectures; training stability

1 Introduction

Missing data are a pervasive problem in statistical analysis and machine learning because most predictive models require complete observations to estimate relationships among variables. Two broad strategies are commonly used to address this issue: case deletion and missing value imputation. Case deletion removes observations containing at least one missing entry, which can substantially reduce sample size and statistical power when missingness is common. In contrast, imputation retains partially observed cases by replacing missing entries with plausible values, but poor imputation can distort variable relationships and introduce bias.

Rubin [1] distinguished three missing-data mechanisms: Missing Completely At Random (MCAR), Missing At Random (MAR), and Missing Not At Random (MNAR). Under MCAR, missingness is unrelated to either observed or unobserved data. Under MAR, missingness depends on observed variables but not on the missing values themselves. Under MNAR, missingness depends on the unobserved values and is therefore the most difficult setting to handle appropriately.

Median imputation is a simple and widely used baseline because it preserves a measure of central tendency and is relatively robust to outliers [2]. However, it often underestimates variability and weakens multivariate structure. More advanced methods include Multiple Imputation by Chained Equations (MICE), which iteratively models each variable with missingness as a function of the others [3]. MICE is flexible and widely used, but it can be computationally demanding and may perform poorly when relationships are strongly nonlinear or when categorical variables are not modeled appropriately [4,5]. MissForest is another popular method that uses random forests to capture complex nonlinear interactions [6]. Although it performs well for mixed-type data, it can be computationally expensive on large datasets and may degrade under high missingness or highly skewed distributions [7].

Generative adversarial networks (GANs), introduced by Goodfellow et al. [8], established the adversarial learning framework later adapted for missing value imputation [8]. Generative Adversarial Imputation Networks (GAIN) introduced a deep learning framework for imputing missing values through adversarial training [9]. Prior comparisons have shown that GAIN can outperform conventional methods such as MICE and MissForest in several benchmark settings, particularly when missingness is substantial [10–12]. Nevertheless, its performance remains sensitive to data characteristics, including variable type, dimensionality, sample size, and missingness pattern.

Since its introduction, several GAIN-based variants have been proposed, including LFM-D2GAIN [13], GAGIN [14], ClueGAIN [15], ccGAN [16], LWGAIN [17], and MGAIN [18]. These variants illustrate how the GAIN framework has been extended to improve performance in specialized settings and data structures. Publicly available variants also include scGAIN, which was developed for imputing missing gene expression values in single-cell RNA sequencing data [19]; CGAIN, a class-aware extension based on conditional adversarial imputation [20]; and COGAIN, which incorporates weighted loss functions for mixed-type clinical data with substantial missingness and variable imbalance [10]. Related GAN-based approaches have also been proposed beyond the GAIN family, including the GAN-based method of Baro & Borah [21], which addresses missing-data imputation together with class imbalance [21]. Despite this growing literature, practical use of GAIN still faces two important challenges. First, the original implementation depends on the outdated `TensorFlow 1.x` API and several nonstandard utility functions, which reduces accessibility and reproducibility. Second, training is highly sensitive to hyperparameters and can exhibit convergence instability, particularly when missingness is concentrated in a subset of variables [18,22]. Sun et al. [12] likewise noted the lack of standardized software support for GAIN.

To address these limitations, we propose Enhanced Generative Adversarial Imputation Networks (EGAIN), a modernized extension of GAIN implemented in `TensorFlow 2.x`. EGAIN introduces a convolution-based generator and discriminator together with a channel-stacked representation of the input data and mask. In our framework, these two design choices are linked: the aligned data–mask representation allows convolutional filters to learn localized interactions between feature values and missingness indicators. EGAIN also incorporates model checkpointing and training diagnostics for improved stability and usability.

Rather than redefining the underlying adversarial imputation framework, these modifications are intended to improve reproducibility, reduce convergence failures, and strengthen empirical performance, particularly in structured tabular settings. To improve accessibility, we also provide the EGAIN project page on GitHub (https://github.com/asaghafi/EGAIN), which includes installation and usage instructions. We note that EGAIN in this study refers specifically to the missing-value imputation package described here. The empirical evaluation in this study is intentionally focused on direct comparison with the original GAIN implementation and a simple baseline imputation method, since the main objective is to assess whether the proposed modifications improve the robustness, usability, and performance of GAIN itself. In this sense, the motivation for EGAIN is practical as well as predictive: even moderate but consistent improvements may

be meaningful when accompanied by greater stability and easier software adoption. At the same time, the proposed modifications are motivated primarily by architectural and implementation considerations, and the present study does not claim a formal theoretical derivation of their advantage. The proposed model is evaluated on benchmark datasets under multiple MCAR configurations and compared with the original GAIN implementation and median imputation.

2 Model Description

Generative Adversarial Imputation Networks (GAIN), introduced by Yoon et al. [9], formulate missing value imputation within a generative adversarial framework. The method uses a generator (G) to estimate missing entries and a discriminator (D) to distinguish observed values from imputed values. The generator produces

$$\hat{X} = G\left(\tilde{X}, M, Z\right), \tag{1}$$

where $\tilde{X}$ denotes the incomplete data matrix with missing entries initially filled with zeros, M is the binary mask matrix with entries equal to 1 for observed values and 0 for missing values, and Z is random noise injected only at missing positions.

After the generator (G) imputes the missing values, the discriminator (D) receives the imputed data together with a hint matrix H that partially reveals the missingness pattern and outputs

$$D\left(\hat{X}, H\right), \tag{2}$$

which estimates, for each component, the probability that the value is observed rather than imputed. Training proceeds by optimizing opposing loss functions for the generator and discriminator. The discriminator is trained to maximize classification (real/imputed) accuracy by minimizing the binary cross-entropy objective

$$\mathcal{L}_D = -\mathbb{E}_{\hat{X},M,H}\left[M \log D(\hat{X}, H) + (1 - M) \log\left(1 - D(\hat{X}, H)\right)\right]. \tag{3}$$

The generator is trained to reduce the discriminator's ability to identify imputed entries:

$$\mathcal{L}_G = -\mathbb{E}_{\tilde{X},M}\left[(1 - M) \log D(\hat{X}, H)\right]. \tag{4}$$

This loss function is applied only to imputed missing entries ($m_i = 0$) and penalizes the generator (G) when the discriminator (D) correctly assigns a low probability to those entries being observed rather than imputed. To encourage the generator to produce realistic values that deceive the discriminator, a reconstruction loss is added to the generator:

$$\mathcal{L}_M = \begin{cases} \sum m_i (x_i - \hat{x}_i)^2, & \text{if } x_i \text{ is continuous,} \\ -\sum m_i (x_i \log(\hat{x}_i)), & \text{if } x_i \text{ is binary.} \end{cases} \tag{5}$$

This loss function is only applied to observed values ($m_i = 1$). The total generator loss is therefore

$$\mathcal{L}_G^{total} = \mathcal{L}_G + \alpha \mathcal{L}_M, \tag{6}$$

where α controls the relative contribution of the reconstruction term. In practice, the original GAIN implementation and many of its successors mainly use the continuous reconstruction component.

EGAIN preserves the basic adversarial structure of GAIN while introducing several modifications intended to improve stability, usability, and performance:

- **TensorFlow 2.x implementation.** EGAIN is implemented using the TensorFlow 2.x API rather than the legacy TensorFlow 1.x framework. This improves compatibility with current software environments and simplifies maintenance and reproducibility [23].
- **Standardized built-in functions.** Several custom utility routines in the original implementation are replaced with standard TensorFlow/Keras functionality for initialization, scaling, and model construction, reducing software complexity and debugging burden.
- **Convolution-based generator and discriminator.** Instead of relying only on dense layers, EGAIN uses convolutional layers in both adversarial components. This choice was motivated by the way the input is represented in EGAIN rather than by an assumption that convolution is universally preferable for tabular data. Specifically, once the feature values and their corresponding missingness indicators are arranged as aligned channels, convolution provides a structured way to learn local patterns involving neighboring features and the interaction between observed and missing entries.
- **Channel-stacked input representation.** Rather than concatenating the data matrix and mask vector into a single flat input, EGAIN treats them as separate but aligned channels. This representation preserves the pairing between each feature value and its missingness indicator and makes it possible for convolutional filters to operate jointly on both sources of information. In this sense, the use of convolution in EGAIN is tied directly to the proposed data–mask arrangement.
- **Checkpointing and convergence diagnostics.** EGAIN stores the best-performing model weights during training and provides generator/discriminator loss plots. These diagnostics help identify unstable runs and support more informed hyperparameter selection.
- **Improved support for hyperparameter tuning.** By exposing training behavior through diagnostic curves, EGAIN reduces the trial-and-error burden associated with selecting parameters such as batch size and the generator reconstruction weight α.
- **Reproducible software packaging.** EGAIN is distributed as a documented Python package with example workflows, facilitating reuse and replication across applications.

3 Datasets and Experimental Setup

Table 1 summarizes the benchmark datasets used to compare EGAIN with the original GAIN implementation [9] and baseline median imputation [2]. All datasets listed in Table 1 were obtained from the UCI Machine Learning Repository [24]. The comparison was designed to evaluate EGAIN primarily as a direct extension of GAIN rather than as a comprehensive benchmark against all available imputation methods. For that reason, the experiments focused on the original GAIN model and a simple baseline imputation approach. For each dataset, we evaluated multiple missingness settings across a range of missing-data rates. Missing values were generated under the MCAR mechanism following Rubin [1], with different random seeds used to create multiple incomplete versions of each dataset.

Table 1: Summary of the benchmark datasets used in this study, all obtained from the UCI machine learning repository [24].

Dataset	Cases (n)	Predictors (d)	Description
Breast Cancer Wisconsin	569	30	30 numerical
Spambase	4601	57	57 numerical
Letter Recognition	20,000	16	16 categorical
Default of Credit Card Client	30,000	23	14 numerical, 9 binary categorical
Online News Popularity	39,797	58	44 numerical, 14 binary categorical

Imputation was then performed on each incomplete dataset using either EGAIN or GAIN, without access to the original complete data during the imputation step. Hyperparameter selection combined systematic search and practical calibration. Specifically, the number of training iterations was selected through grid search separately for GAIN and EGAIN, whereas other parameters were chosen using a combination of cross-validation and diagnostic inspection of training behavior. To maintain direct comparability with the original GAIN study and related GAIN-based literature, performance was evaluated by comparing imputed entries with their true values from the original complete dataset using root mean squared error (RMSE). RMSE was selected because it provides a standard summary measure of imputation error magnitude and penalizes larger deviations from the true values more strongly, making it a widely used metric in imputation research:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}, \tag{7}$$

where n denotes the number of imputed entries, x_i is the true value, and $\hat{x}_i$ is the corresponding imputed value. This choice is also consistent with several published studies on GAIN-based imputation methods that use RMSE or MSE as the primary performance criterion [9,13,14,16,20].

To improve the stability of the comparison, the full procedure was repeated 25 times using the same random seeds for EGAIN and GAIN. All variables were min–max scaled to the interval [0, 1] before RMSE computation so that variables measured on different scales contributed comparably to the error metric. The hint rate was fixed at 90% for all experiments, consistent with the original GAIN study. Because there is limited prior guidance on systematic optimization of this parameter, the same value was used across all runs. Batch size was selected according to dataset size while keeping the batch-size-to-sample-size ratio below 10% of the total number of cases. Specifically, batch sizes of 32 or 64 were used for small datasets (fewer than 1,000 cases), 64 or 128 for medium datasets (1,000–5,000 cases), and 128 or 256 for large datasets (more than 5,000 cases). This choice provided reliable performance while controlling runtime. The hyperparameter α was selected by examining generator and discriminator loss values and choosing settings that yielded stable and interpretable training behavior. Accordingly, this step should be viewed as a practical calibration procedure rather than a formal optimization strategy. Reported line plots summarize the mean RMSE and standard deviation across the 25 runs. Overall differences in RMSE among the imputation methods were first evaluated using analysis of variance (ANOVA). When the overall ANOVA was significant, pairwise differences were then assessed using Tukey's Honestly Significant Difference (HSD) multiple-comparisons test. Additional implementation details and model-specific hyperparameters are provided in the supplementary materials.

4 Results

Table 2 provides an overall summary of imputation performance across all benchmark datasets. EGAIN consistently outperformed the original GAIN implementation, achieving statistically significant RMSE reductions of 3.11%, 19.85%, 15.12%, 16.48%, and 1.61% on the Breast Cancer, Spambase, Letter Recognition, Credit Card Client, and News Popularity datasets, respectively. Additionally, EGAIN consistently surpassed the baseline Median imputation method across all datasets, whereas GAIN failed to do so in both the Spambase and Letter Recognition tasks. Beyond improved accuracy, one of EGAIN's most important contributions is its resolution of a key limitation in GAIN—training instability. Across 2000 simulation runs, GAIN failed to complete imputations in approximately 39% of cases due to convergence issues. In contrast, EGAIN completed all runs successfully.

Table 2: Average RMSE with standard deviations (in parentheses) across all experiments for different imputation methods and datasets. Boldface values indicate the smallest average RMSE among the compared imputation methods for each dataset.

Dataset	Median	GAIN	EGAIN
Breast Cancer Wisconsin	0.0526 (0.0358)	0.0385 (0.0331)	**0.0373** (0.0314)
Spambase	0.0173 (0.0153)	0.0200 (0.0157)	**0.0161** (0.0151)
Letter Recognition	0.0666 (0.0354)	0.0735 (0.0434)	**0.0624** (0.0357)
Default of Credit Card Client	0.0417 (0.0264)	0.0413 (0.0284)	**0.0345** (0.0263)
Online News Popularity	0.0637 (0.0558)	0.0592 (0.0548)	**0.0582** (0.0525)

Taken together, the results show that EGAIN was generally more accurate and more stable than the original GAIN implementation, with the clearest advantages appearing under more challenging or structured missingness settings. Detailed run-level results, including examples of failed GAIN imputations, are provided in the supplementary materials.

Fig. 1 (left) and Table 3 summarize the RMSE performance of GAIN, EGAIN, and median imputation on the Breast Cancer dataset when missing values were introduced completely at random across the numerical predictors. After a significant overall ANOVA, Tukey's HSD multiple-comparisons test showed that EGAIN significantly outperformed GAIN across all missingness levels ($p < 0.001$). Both adversarial methods generally performed better than median imputation, although this advantage narrowed as missingness increased. At the highest missingness level, GAIN fell below the median baseline, whereas EGAIN maintained better overall performance. GAIN was also more sensitive to the number of training iterations and showed reduced convergence stability.

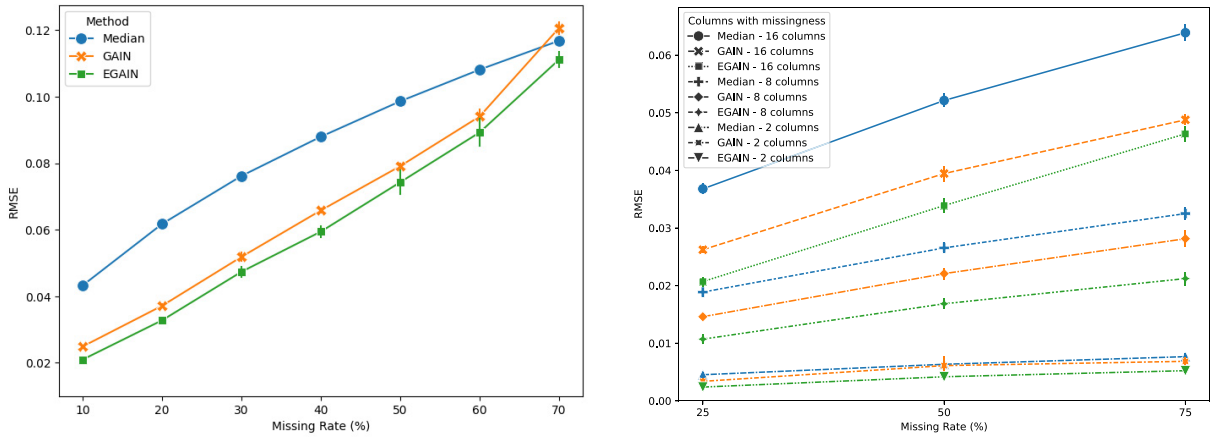

Figure 1: Performance comparison of GAIN and EGAIN on the breast cancer dataset: (**left**) MCAR across all numerical predictors; (**right**) MCAR restricted to randomly selected subsets of numerical predictors.

The differences between the two adversarial models were even more apparent when missingness was restricted to a subset of columns (Fig. 1, right; Table 4). Under this setting, after a significant overall ANOVA, Tukey's HSD multiple-comparisons test again showed that EGAIN outperformed GAIN across all tested conditions ($p < 0.001$). Both adversarial methods remained better than median imputation, but GAIN showed greater sensitivity to training length and more frequent instability.

Table 3: Average RMSE with standard deviations in parentheses across 25 runs by imputation method, dataset, and missing data rate. Boldface values indicate the smallest average RMSE within each dataset and missing data rate.

Dataset	Missing Rate Imputation	10%	20%	30%	40%	50%	60%	70%
Breast Cancer Wisconsin	EGAIN	**0.0210** (0.0005)	**0.0328** (0.0011)	**0.0474** (0.0037)	**0.0594** (0.0039)	**0.0743** (0.0087)	**0.0893** (0.0102)	**0.1111** (0.0056)
	GAIN	0.0249 (0.0007)	0.0372 (0.0010)	0.0519 (0.0021)	0.0658 (0.0010)	0.0791 (0.0014)	0.0940 (0.0034)	0.1206 (0.0034)
	Median	0.0433 (0.0014)	0.0618 (0.0014)	0.0762 (0.0014)	0.0880 (0.0011)	0.0986 (0.0010)	0.1081 (0.0009)	0.1168 (0.0009)
Spambase	EGAIN	**0.0147** (0.0004)	**0.0218** (0.0001)	**0.0278** (0.0000)	**0.0324** (0.0000)	**0.0368** (0.0000)	**0.0414** (0.0002)	**0.0441** (0.0001)
	GAIN	0.0148 (0.0002)	0.0219 (0.0004)	0.0279 (0.0005)	0.0326 (0.0004)	0.0373 (0.0002)	0.0416 (0.0002)	0.0457 (0.0001)
	Median	0.0165 (0.0005)	0.0239 (0.0006)	0.0294 (0.0007)	0.0341 (0.0006)	0.0382 (0.0005)	0.0419 (0.0004)	0.0453 (0.0004)
Letter Recognition	EGAIN	0.0399 (0.0012)	0.0575 (0.0014)	0.0728 (0.0019)	**0.0860** (0.0026)	**0.0993** (0.0032)	**0.1111** (0.0024)	**0.1199** (0.0016)
	GAIN	**0.0395** (0.0011)	**0.0559** (0.0013)	**0.0703** (0.0011)	0.0861 (0.0008)	0.1003 (0.0023)	0.1177 (0.0038)	0.1460 (0.0090)
	Median	0.0457 (0.0003)	0.0646 (0.0002)	0.0791 (0.0003)	0.0914 (0.0002)	0.1021 (0.0002)	0.1120 (0.0003)	0.1210 (0.0003)
Default of Credit Card Client	EGAIN	**0.0303** (0.0015)	**0.0442** (0.0018)	**0.0550** (0.0015)	**0.0632** (0.0013)	**0.0704** (0.0009)	**0.0767** (0.0017)	**0.0826** (0.0013)
	GAIN	0.0304 (0.0009)	0.0455 (0.0012)	0.0585 (0.0018)	0.0689 (0.0015)	0.0787 (0.0010)	0.0859 (0.0024)	0.0929 (0.0014)
	Median	0.0330 (0.0001)	0.0467 (0.0002)	0.0572 (0.0002)	0.0661 (0.0002)	0.0738 (0.0002)	0.0809 (0.0001)	0.0874 (0.0001)
Online News Popularity	EGAIN	0.0552 (0.0012)	0.0787 (0.0019)	0.0988 (0.0015)	**0.1158** (0.0014)	**0.1300** (0.0014)	**0.1431** (0.0014)	**0.1561** (0.0041)
	GAIN	**0.0538** (0.0007)	**0.0772** (0.0008)	**0.0980** (0.0008)	0.1182 (0.0007)	0.1337 (0.0005)	0.1475 (0.0009)	0.1625 (0.0008)
	Median	0.0624 (0.0002)	0.0883 (0.0002)	0.1081 (0.0002)	0.1249 (0.0002)	0.1396 (0.0002)	0.1530 (0.0002)	0.1652 (0.0002)

Table 4: Average RMSE with standard deviations in parentheses across multiple runs by dataset and imputation method. Columns represent missing data rates and number of missing columns. Boldface values indicate the smallest average RMSE within each dataset, missing data rate, and missing-column setting.

	Columns	2			8			14/16*		
Data	Missing Rate / Method	25%	50%	75%	25%	50%	75%	25%	50%	75%
Breast Cancer Wisconsin	EGAIN	**0.0024** (0.0003)	**0.0042** (0.0004)	**0.0052** (0.0004)	**0.0107** (0.0016)	**0.0169** (0.0019)	**0.0212** (0.0026)	**0.0207** (0.0018)	**0.0339** (0.0029)	**0.0463** (0.0031)
	GAIN	0.0034 (0.0009)	0.0061 (0.0027)	0.0069 (0.0011)	0.0146 (0.0005)	0.0221 (0.0016)	0.0281 (0.0022)	0.0262 (0.0007)	0.0395 (0.0017)	0.0488 (0.0014)
	Median	0.0045 (0.0006)	0.0063 (0.0009)	0.0077 (0.0012)	0.0189 (0.0016)	0.0265 (0.0020)	0.0325 (0.0024)	0.0368 (0.0020)	0.0521 (0.0026)	0.0639 (0.0032)
Spambase	EGAIN	**0.0010** (0.0000)	**0.0014** (0.0000)	**0.0016** (0.0000)	**0.0025** (0.0000)	**0.0038** (0.0000)	**0.0047** (0.0000)	**0.0052** (0.0001)	**0.0079** (0.0000)	**0.0099** (0.0001)
	GAIN	0.0010 (0.0000)	0.0015 (0.0001)	0.0020 (0.0006)	0.0027 (0.0002)	0.0041 (0.0003)	0.0052 (0.0004)	0.0054 (0.0001)	0.0081 (0.0002)	0.0100 (0.0001)
	Median	0.0010 (0.0004)	0.0014 (0.0005)	0.0017 (0.0005)	0.0035 (0.0007)	0.0050 (0.0009)	0.0062 (0.0011)	0.0070 (0.0008)	0.0101 (0.0010)	0.0124 (0.0013)
Letter Recognition	EGAIN	**0.0075** (0.0012)	**0.0101** (0.0017)	**0.0127** (0.0023)	**0.0312** (0.0024)	**0.0453** (0.0027)	**0.0565** (0.0045)	**0.0563** (0.0023)	**0.0855** (0.0023)	**0.1067** (0.0041)
	GAIN	0.0085 (0.0009)	0.0118 (0.0015)	0.0160 (0.0020)	0.0330 (0.0025)	0.0483 (0.0010)	0.0690 (0.0042)	0.0567 (0.0020)	0.0865 (0.0027)	0.1532 (0.0141)
	Median	0.0088 (0.0008)	0.0123 (0.0013)	0.0150 (0.0016)	0.0364 (0.0013)	0.0515 (0.0018)	0.0631 (0.0022)	0.0632 (0.0011)	0.0893 (0.0015)	0.1094 (0.0019)
Default of Credit Card Client	EGAIN	**0.0038** (0.0001)	**0.0051** (0.0001)	**0.0063** (0.0002)	**0.0111** (0.0002)	**0.0155** (0.0002)	**0.0193** (0.0005)	**0.0163** (0.0004)	**0.0231** (0.0004)	**0.0286** (0.0005)
	GAIN	0.0043 (0.0004)	0.0064 (0.0011)	0.0083 (0.0020)	0.0130 (0.0007)	0.0190 (0.0013)	0.0240 (0.0027)	0.0196 (0.0008)	0.0283 (0.0011)	0.0338 (0.0016)
	Median	0.0045 (0.0036)	0.0064 (0.0051)	0.0078 (0.0062)	0.0180 (0.0058)	0.0255 (0.0082)	0.0312 (0.0101)	0.0311 (0.0064)	0.0441 (0.0091)	0.0540 (0.0111)
Online News Popularity	EGAIN	**0.0027** (0.0015)	**0.0039** (0.0022)	**0.0048** (0.0027)	**0.0114** (0.0025)	**0.0164** (0.0033)	**0.0200** (0.0044)	**0.0224** (0.0035)	**0.0328** (0.0050)	**0.0397** (0.0066)
	GAIN	0.0029 (0.0015)	0.0047 (0.0022)	0.0062 (0.0030)	0.0132 (0.0031)	0.0171 (0.0042)	0.0215 (0.0049)	0.0256 (0.0045)	0.0349 (0.0035)	0.0433 (0.0076)
	Median	0.0032 (0.0019)	0.0045 (0.0027)	0.0055 (0.0033)	0.0134 (0.0030)	0.0190 (0.0042)	0.0232 (0.0052)	0.0263 (0.0043)	0.0372 (0.0061)	0.0456 (0.0074)

Note: *14 for letter recognition, default of credit card client; 16 for breast cancer wisconsin, spambase, online news popularity.

Fig. 2 shows the discriminator and generator loss curves for EGAIN during training on the Breast Cancer dataset. The plot illustrates how the two adversarial components evolve over training and provides practical guidance for selecting α so that the initial loss scales are comparable. Around iteration 100, the generator loss $\mathcal{L}_G$ begins to increase slightly, reflecting the discriminator's improved ability to distinguish real from imputed values, thereby pushing the generator to produce more realistic imputations. This plot also facilitates the selection of the hyperparameter α by enabling alignment of the initial loss scales between the discriminator and generator. In this example, $\alpha = 80$ produced a suitable balance. Although training continued for 1000 iterations, the minimum generator loss occurred near iteration 520, at which point EGAIN stored the corresponding model weights through checkpointing. This feature is important because it reduces the impact of overtraining and improves robustness relative to the original GAIN implementation. For visualization purposes, the discriminator loss was multiplied by 10 to improve readability of the joint loss plot.

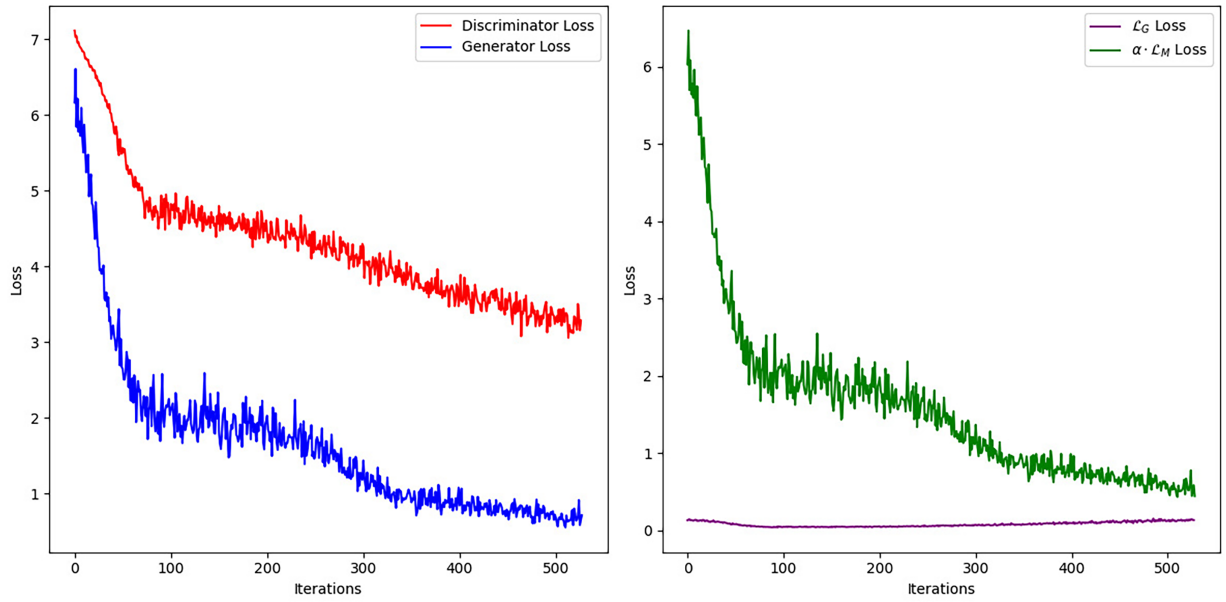

Figure 2: EGAIN training losses on the breast cancer dataset: generator and discriminator losses (**left**) and generator loss components (**right**).

Fig. 3 presents the results for the Spambase dataset. EGAIN achieved the lowest RMSE across the evaluated missingness levels, and the overall difference among methods was significant according to ANOVA ($p < 0.001$). Its advantage became more apparent when missingness was confined to subsets of predictors. As in the Breast Cancer dataset, GAIN was less stable and more sensitive to training iterations.

Fig. 4 shows the results for the Letter Recognition dataset. EGAIN consistently outperformed median imputation. GAIN showed slightly lower RMSE at lower missingness levels, but EGAIN became more competitive and ultimately more robust as missingness increased. When missingness was confined to subsets of predictors, EGAIN showed a clearer advantage over both comparison methods.

Fig. 5 presents the results for the Default of Credit Card Client dataset. EGAIN consistently outperformed both GAIN and median imputation across the evaluated settings. Its advantage was especially pronounced when missingness was confined to subsets of predictors. As in the earlier datasets, GAIN remained highly sensitive to the number of training iterations and was less stable during training.

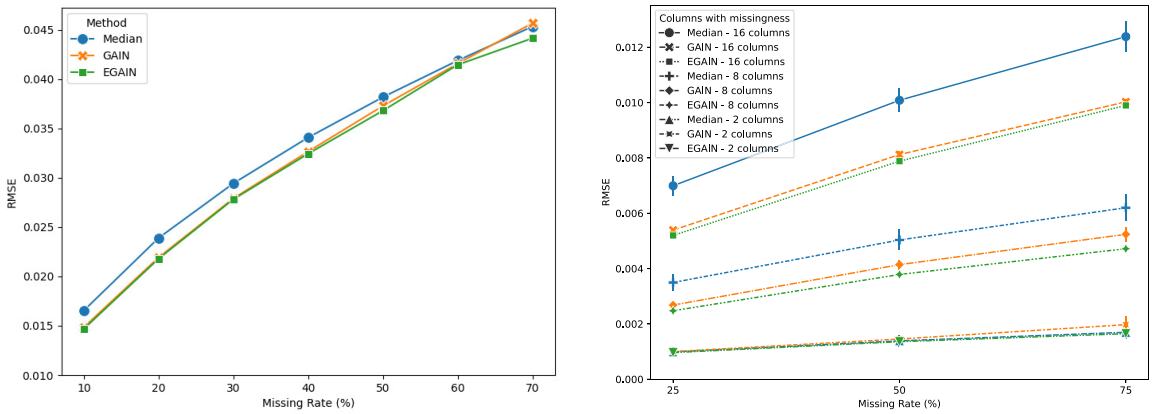

Figure 3: Performance comparison of GAIN and EGAIN on the spambase dataset: (**left**) MCAR across all numerical predictors; (**right**) MCAR restricted to randomly selected subsets of predictors.

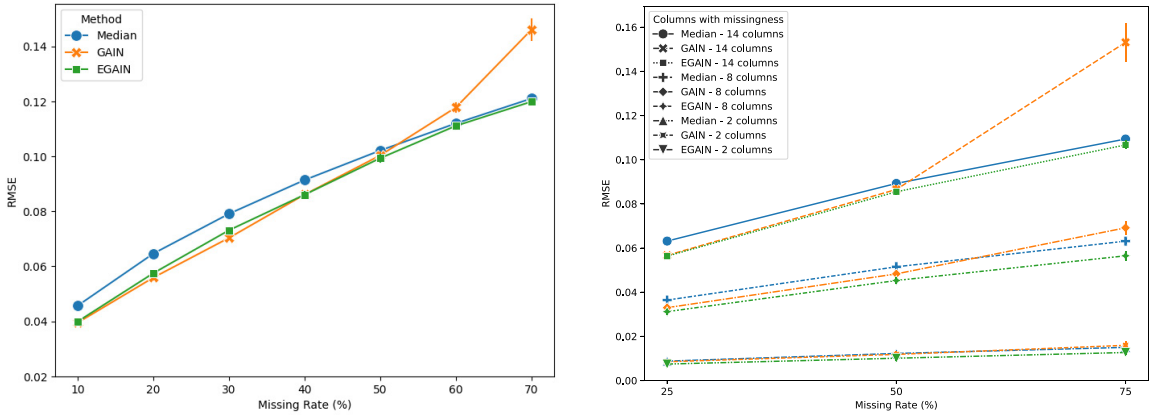

Figure 4: Performance comparison of GAIN and EGAIN on the letter recognition dataset: (**left**) MCAR across all categorical predictors; (**right**) MCAR restricted to randomly selected subsets of categorical predictors.

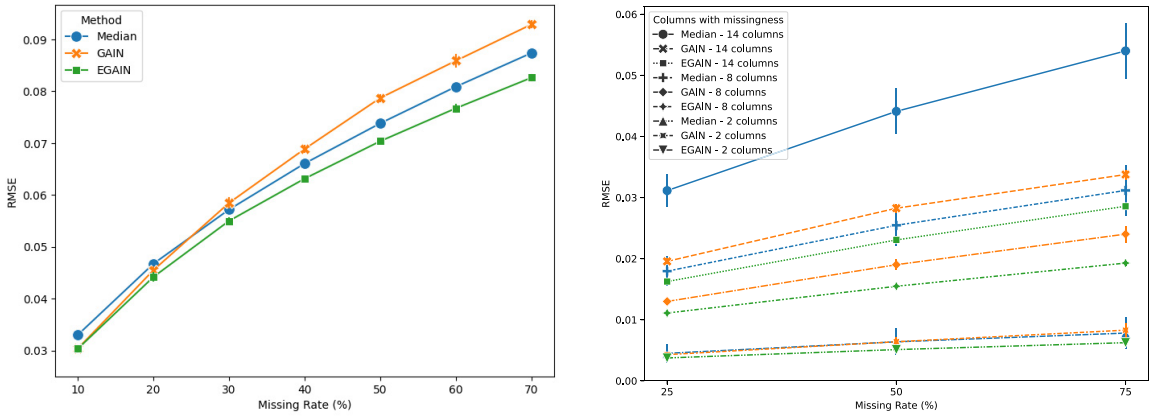

Figure 5: Performance comparison of GAIN and EGAIN on the default of credit card client dataset: (**left**) MCAR across all numerical predictors; (**right**) MCAR restricted to randomly selected subsets of numerical predictors.

Fig. 6 summarizes the results for the Online News Popularity dataset. Both GAIN and EGAIN outperformed median imputation when missingness was introduced across all predictors. GAIN performed slightly better at lower missingness levels, whereas EGAIN became superior as missingness increased. Under subset-based missingness, EGAIN again showed stronger and more consistent performance, indicating improved robustness under structured missingness.

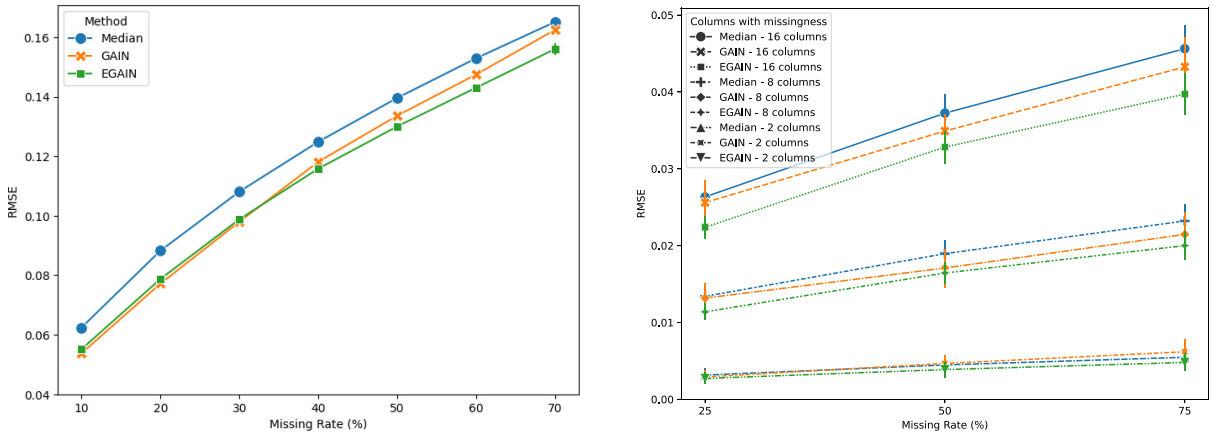

Figure 6: Performance comparison of GAIN and EGAIN on the online news popularity dataset: (**left**) MCAR across all predictors; (**right**) MCAR restricted to randomly selected subsets of predictors.

5 Discussion

This study revisited GAIN from both methodological and software perspectives. Although GAIN remains one of the most influential deep-learning approaches for missing value imputation, its practical use is limited by implementation obsolescence, sensitivity to hyperparameter selection, and convergence instability. The findings of this study suggest that these issues are not merely technical inconveniences; rather, they directly affect the reliability and consistency of imputation results across datasets and missingness settings.

EGAIN was developed to address these limitations through a TensorFlow 2.x implementation, standardized software components, checkpoint-based model selection, diagnostic loss monitoring, and a convolution-based architecture applied to channel-stacked data and mask representations. Across the benchmark datasets examined in this study, EGAIN generally achieved lower RMSE than the original GAIN implementation and showed greater robustness when missingness was concentrated in a subset of variables. Although the numerical gains were generally moderate, they were consistent across a range of settings and should be interpreted together with the improved convergence behavior of the method. This setting is particularly relevant in practice because missingness in real-world datasets is often structured rather than uniformly distributed across all features.

A key architectural distinction of EGAIN is the replacement of purely dense adversarial networks with convolutional layers. Although convolutional models are most commonly associated with spatial data, prior studies have shown that they can also be beneficial for tabular data when local dependencies or structured feature interactions exist. In the present study, the motivation for convolution was not that tabular data are inherently spatial, but that the proposed channel-stacked representation of the data matrix and missingness mask creates a structured local arrangement that convolutional filters can exploit. By operating jointly on aligned feature values and missingness indicators, the filters can learn patterns that may be less explicit in a fully dense architecture. The present results are consistent with the view that this representation improves imputation quality, particularly under structured missingness scenarios. However, this rationale is

architectural and empirical rather than fully theoretical. The current study does not provide a formal analysis of why the proposed representation and training modifications should improve convergence or imputation accuracy, and future work is needed to develop a stronger theoretical account of these effects in addition to ablation-based empirical validation.

Another important contribution of EGAIN lies in its practical training workflow. The integration of checkpointing and loss-curve diagnostics makes the model easier to monitor and less dependent on selecting a single arbitrary training iteration. In contrast to the original GAIN implementation, which may deteriorate when training continues beyond a suitable range, EGAIN allows restoration of the best-performing model state observed during training. This practical advantage matters because the contribution of EGAIN is not limited to RMSE reduction alone; it also includes improved training stability, reproducibility, and ease of use relative to the original implementation. Empirically, this difference was substantial: across 2000 simulation runs, GAIN failed to complete imputations in approximately 39% of cases, whereas EGAIN completed all runs successfully. These findings support the claim that the proposed framework is more stable in practice, although the present study does not provide a formal theoretical explanation of how each design modification contributes to that stability.

The improvements introduced in EGAIN come with increased computational cost. For example, on the Default of Credit Card Client dataset, imputing 20% missing values with 1000 iterations and batch size 256 required approximately 20 seconds, which was about three to four times slower than the original GAIN implementation. However, the shorter runtime of GAIN must be interpreted alongside its lower stability, since it more frequently failed to produce usable imputations under the same training conditions. The practical trade-off is therefore not simply speed vs. accuracy, but speed vs. reliability, reproducibility, and usability. In addition, the present comparison should be interpreted in light of the fact that imputation quality was assessed only through RMSE rather than through a broader set of error measures or downstream analytical tasks.

Because EGAIN was evaluated as an integrated framework, the current study does not isolate the individual contribution of each design component through a formal ablation study. However, our experiments indicated that simply upgrading the original GAIN implementation from TensorFlow 1 to TensorFlow 2 did not improve imputation performance. This suggests that the gains observed in EGAIN cannot be attributed solely to software modernization, but instead reflect the broader set of architectural and training modifications introduced in the proposed framework. Moreover, the comparative evaluation was intentionally limited to the original GAIN implementation and a simple baseline method. Accordingly, the present findings support the claim that EGAIN improves upon GAIN under the tested settings, but they should not be interpreted as demonstrating superiority over the broader landscape of imputation methods. Similarly, the hyperparameter selection procedure was only partially systematic, since it combined grid search for the number of training iterations with practical calibration of other parameters based on loss behavior, dataset size, and runtime considerations.

Finally, the practical interpretation of the present findings should be tempered by the fact that all experiments were conducted under MCAR settings, whereas real-world datasets often exhibit MAR or MNAR missingness. For this reason, the current results should be viewed as evidence of improved performance and stability under controlled benchmark conditions rather than as a complete characterization of EGAIN across all missing-data mechanisms.

6 Conclusion

In this paper, we introduced Enhanced Generative Adversarial Imputation Networks (EGAIN), a practical extension of GAIN designed to improve missing value imputation in tabular datasets through a

modernized implementation, a channel-stacked data–mask representation, convolution-based adversarial components, checkpoint-based model selection, and diagnostic training support. Across the benchmark datasets examined in this study, EGAIN generally achieved lower RMSE than the original GAIN implementation and showed greater robustness, particularly when missingness was concentrated in a subset of variables. Although the numerical improvements were often moderate, they were consistent across a range of settings and were accompanied by substantial gains in stability and usability. In particular, EGAIN completed all experimental runs successfully, whereas the original GAIN implementation exhibited frequent convergence-related failures under the same general evaluation framework.

These findings suggest that the proposed modifications improve the practical reliability of adversarial imputation beyond accuracy alone. The main contribution of EGAIN should therefore be understood not as a completely new imputation paradigm, but as an empirically motivated and practically relevant refinement of GAIN that improves reproducibility, accessibility, and training stability while also strengthening imputation performance under the evaluated settings.

This study nevertheless has several limitations. First, the individual contributions of the proposed components were not isolated through a formal ablation study, making it difficult to determine which modifications contributed most strongly to the observed gains. Second, the empirical evaluation was restricted to MCAR settings, so the behavior of EGAIN under MAR and MNAR mechanisms remains to be established. Third, performance was assessed only through RMSE, which, although widely used and appropriate for direct comparison with prior GAIN-based studies, provides only a partial assessment of imputation quality. Fourth, the comparative analysis was intentionally limited to the original GAIN implementation and a simple baseline method, and therefore does not constitute a broad benchmark against alternative imputation approaches such as MICE, MissForest, or more recent GAIN variants. Fifth, the hyperparameter selection process combined grid search for selected parameters with practical calibration of others, rather than relying on a fully systematic optimization framework. Finally, while the revised manuscript provides empirical evidence of improved stability, it does not offer a formal theoretical explanation for why the proposed architectural and training modifications reduce convergence failures.

Future research should therefore pursue several directions. Most importantly, ablation studies are needed to disentangle the contribution of each architectural and training modification. Broader comparative evaluations against established and recent imputation methods would help position EGAIN more clearly within the current literature. Additional experiments under MAR and MNAR settings are necessary to assess practical robustness under more realistic missing-data mechanisms. It would also be valuable to evaluate EGAIN using additional error measures and downstream analytical tasks, including classification, regression, and statistical estimation after imputation. More systematic hyperparameter optimization procedures and deeper theoretical analysis of the proposed representation, architecture, and stability properties would further strengthen understanding of when and why EGAIN is most beneficial.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Methodology: Abolfazl Saghafi; software: Abolfazl Saghafi; validation: Abolfazl Saghafi, Soodeh Moallemian, Miray Budak and Rutvik Deshpande; visualization: Abolfazl Saghafi, Soodeh Moallemian, Miray Budak and Rutvik Deshpande; supervision: Abolfazl Saghafi; writing—original draft: Abolfazl Saghafi; writing—review and editing: Abolfazl Saghafi, Soodeh Moallemian, Miray Budak and Rutvik Deshpande. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used in this study are publicly available from the UCI Machine Learning Repository and are cited in the reference list. Information on the EGAIN package, including installation and usage instructions, is available at https://github.com/asaghafi/EGAIN.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

API	Application programming interface
GAIN	Generative adversarial imputation network
EGAIN	Enhanced generative adversarial imputation network
MAR	Missing at random
MCAR	Missing completely at random
MNAR	Missing not at random
MICE	Multiple imputation by chained equations
LFM-D2GAIN	Latent factor model with dual discriminator GAIN
GAGIN	Generative adversarial guider imputation network
ccGAIN	Conditional clinical GAIN
LWGAIN	Loss wasserstein GAIN
scGAIN	Single-cell GAIN
CGAIN	Conditional GAIN
TF	TensorFlow
RMSE	Root mean squared error

References

1. Rubin D. Inference and missing data. Biometrika. 1976;63(3):581–92. doi:10.1093/biomet/63.3.581.
2. Little RJA, Rubin DB. Statistical analysis with missing data. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2019.
3. Van Buuren S, Groothuis-Oudshoorn K. MICE: Multivariate imputation by chained equations in R. J Stat Softw. 2011;45:1–67.
4. Azur MJ, Stuart EA, Frangakis C, Leaf PJ. Multiple imputation by chained equations: what is it and how does it work? Int J Methods Psychiatr Res. 2011;20(1):40–9.
5. White I, Royston P, Wood A. Multiple imputation using chained equations: Issues and guidance for practice. Stat Med. 2011;30:377–99.
6. Stekhoven D, Buhlmann P. MissForest—non-parametric missing value imputation for mixed-type data. Bioinformatics. 2011;28(1):112–18. doi:10.1093/bioinformatics/btr597.
7. Waljee A, Mukherjee A, Singal A, Zhang Y, Warren J, Balis U, et al. Comparison of imputation methods for missing laboratory data in medicine. BMJ Open. 2013;3(8):e002847. doi:10.1136/bmjopen-2013-002847.
8. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Proceedings of the International Conference on Neural Information Processing Systems; 2014 Dec 8–13; Montreal, QC, Canada.
9. Yoon J, Jordon J, Van Der Schaar M. GAIN: missing data imputation using generative adversarial nets. In: Proceedings of the 35th International Conference on Machine Learning; 2018 Jul 10–15; Stockholm, Sweden.
10. Dong W, Fong D, Yoon J, Wan E, Bedford L, Tang E, et al. Generative adversarial networks for imputing missing data for big data clinical research. BMC Med Res Methodol. 2021;21(1):78. doi:10.1186/s12874-021-01272-3.
11. Shahbazian R, Greco S. Generative adversarial networks assist missing data imputation: A comprehensive survey and evaluation. IEEE Access. 2023;11:88908–28.
12. Sun Y, Li J, Xu Y, Zhang T, Wang X. Deep learning vs. conventional methods for missing data imputation: a review and comparative study. Expert Syst Appl. 2023;227(87):120201. doi:10.1016/j.eswa.2023.120201.

13. Shen Y, Zhang C, Zhang S, Yan J, Bu F. LFM-D2GAIN: an improved missing data imputation method based on generative adversarial imputation nets. In: Proceedings of the 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA); 2022 Feb 25–27; Changchun, China.
14. Wang W, Chai Y, Li Y. GAGIN: generative adversarial guider imputation network for missing data. Neural Comput Appl. 2022;34:7597–610.
15. Zhao S. ClueGAIN: application of transfer learning on generative adversarial imputation nets (GAIN). arXiv:2302.03140. 2023.
16. Bernardini M, Doinychko A, Romeo L, Frontoni M, Amini M. A novel missing data imputation approach based on clinical conditional generative adversarial networks applied to EHR datasets. Comput Biol Med. 2023;163(W1):107188. doi:10.1016/j.compbiomed.2023.107188.
17. Qian H, Geng Y, Wang H, Wu X, Li M. LWGAIN: An improved missing data imputation method based on generative adversarial imputation network. In: Proceedings of the 2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL); 2024 Apr 19–21; Zhuhai, China.
18. Qin X, Shi H, Dong X, Zhang S, Yuan L. Improved generative adversarial imputation networks for missing data. Appl Intell. 2024;54(21):11068–82. doi:10.1007/s10489-024-05814-2.
19. Gunady MK, Kancherla J, Bravo HC, Feizi S. scGAIN: Single cell RNA-seq data imputation using generative adversarial networks. BioArchive. 2019;837302(1):1. doi:10.1101/837302.
20. Awan SE, Bennamoun M, Sohel F, Sanfilippo FM, Dwivedi G. Imputation of missing data with class imbalance using conditional generative adversarial networks. Neurocomputing. 2021;453(4):164–71. doi:10.1016/j.neucom.2021.04.010.
21. Baro P, Borah MD. GAN-based approach for data imputation and handling class imbalance using one class ensemble. Appl Soft Comput. 2025;182(5):113540. doi:10.1016/j.asoc.2025.113540.
22. Kazemi A, Meidani H. IGANI: iterative generative adversarial networks for imputation with application to traffic data. arXiv:2008.04847. 2020.
23. TensorFlow 1.x vs. TensorFlow 2 - Behaviors and APIs. [cited 2026 Mar 1]. Available from: https://www.tensorflow.org/guide/migrate/tf1.
24. The UCI Machine Learning Repository. [cited 2026 Mar 1]. Available from: https://archive.ics.uci.edu.