A Deep Learning Architecture for Combining and Imputing Heterogeneous Metabolomics Datasets

Sadi Celik, Baris Can, Mehmet Ali Erdogan, and Ali Cakmak*
Istanbul Technical University, 34467 Maslak, Istanbul, Turkey.
*Corresponding author: A. Cakmak, email: ali.cakmak@itu.edu.tr

Abstract—Public metabolomics databases offer a large number of datasets. Combined analysis of these data sets may better capture complex molecular mechanisms in diseases. However, most datasets include measurements for only a very small fraction of the known metabolites. Hence, simply putting together these studies leads to very sparse datasets, which do not lend themselves well to training machine learning models. In this paper, we propose two novel approaches for dataset merging and imputation model training: (i) Iterative Similarity-based Merging generates an optimal merge set for each dataset and makes sure that a minimum sparsity threshold is maintained, and (ii) Model-guided Agglomerative Merging combines datasets in pairs to create a single large dataset in an attempt to effectively combine diverse metabolomics datasets while minimizing the likelihood of gaps created by non-overlapping metabolites. In both approaches, after creating joint datasets, Variational Autoencoders (VAE) are employed for imputation model training. We evaluate our approach on the entire set of public datasets from the Metabolomics Workbench. Our results demonstrate that the proposed approaches achieve significantly better imputation performance than the state-of-the-art approach.

Index Terms—Metabolomics, Deep Learning, Missing Data Imputation, Data Merging

1 Introduction

The high-throughput molecular measurement datasets are commonly called "omics" datasets (e.g., genomics, transcriptomics, proteomics, metabolomics, etc.). Among the main omics areas, metabolomics is the closest to the phenotype. Metabolic function is assessed by measuring the relative abundance levels of metabolites. Metabolites are small chemicals that are produced and consumed by various biochemical reactions. Perturbations outside the normal range indicate disease conditions. Multiple combinations of mass spectrometry (MS) in conjunction with liquid chromatography (LC), gas chromatography (GC), and nuclear magnetic resonance (NMR) methods are routinely employed to quantify metabolites. However, this diversity in experimental platforms and subsequent data processing workflows introduces a severe challenge for cross-study harmonization. Because each analytical technique possesses distinct sensitivities and dynamic ranges, the resulting data types are highly heterogeneous. While some targeted studies provide absolute quantification in standard molar units, the vast majority of untargeted studies report relative abundances in arbitrary units. Consequently, a naive combination of these datasets is confounded by disparate numerical scales and inherent batch effects. Any robust dataset merging strategy must therefore look beyond simple feature overlap and account for these varying measurement scales, a challenge we explicitly address in this work through a novel weighted Jaccard similarity metric. During the process, missing metabolite measurements in datasets can occur and affect up to 80% of the variables, contributing to a misinterpretation of the analysis [16].

Besides, publicly available datasets are quite heterogeneous with vastly differing sets of measured metabolites. Hence, simply putting them together leads to sparse datasets (around 97% in Metabolomics Workbench datasets) with huge numbers of missing items [32].

The literature contains two umbrella categories of approaches to solve the metabolomics dataset imputation problem. The first category involves traditional general-purpose statistical model-based and machine learning-based imputation methods, such as k-nearest-neighbor (KNN) imputation [34], random forest (RF) imputation [30], multivariate imputation by chained equations (MICE) [36], etc. are frequently employed ([5], [6], [38], [1], [9]). Xu *et al.* [39] introduced a dimensionality reduction-based, non-negative matrix factorization (NMF) feature extraction technique for MS-based metabolomics imputation. Kumar *et al.* [22] developed an outlier-robust missing values imputation (ORI) technique that combines singular value decomposition (SVD) with an extra layer for replacing outliers. The first category of approaches cannot handle vastly differing sets of metabolites among heterogeneous metabolomics datasets as they capture patterns among metabolites included in a particular dataset. Hence, they work locally on a single metabolomics dataset. Moreover, a recent study by Krutkin et al. [21] demonstrates that even these single-dataset methods should be applied with caution: imputation accuracy degrades substantially when the proportion of missing values is high or when the missingness mechanism (MCAR vs. MNAR) is unknown, underscoring the need for more robust and context-aware imputation frameworks.

The second major category of approaches in the literature involves generative model-based imputation frame-

works ([13], [43], [15]). In comparison to the former category of approaches, the latter approaches are able to learn from multiple metabolomics datasets. Hence, they better generalize owing to their more diverse pattern base. As another advantage, the second category of approaches trains a single model and employs it for all future imputation tasks. However, as a significant drawback, the existing studies in this category assume that all datasets homogeneously measure the same or highly overlapping sets of metabolites. In practice, this is a highly compelling limitation, as the datasets in public metabolomics data repositories (e.g., Metabolomics Workbench [32], Metabolights [18], etc.) are contributed by many different groups with various metabolite sets of interest. Hence, the second category of approaches' use cases are restricted to particular experimental settings.

In order to fill the above gap, in this paper, we propose a novel generative model-based imputation framework that can utilize diverse metabolomics datasets with different metabolite sets to create a single imputation model. In particular, we propose two novel merging strategies: (i) *Iterative Similarity-based Merging* generates an optimal merge set for each dataset and makes sure that a minimum sparsity threshold is maintained, and (ii) *Model-guided Agglomerative Merging* combines datasets in pairs to create a single large dataset in an attempt to effectively combine diverse metabolomics datasets while minimizing the likelihood of gaps created by non-overlapping metabolites. In both approaches, after creating joint datasets, Variational Autoencoders (VAE) [20] are employed for model training. VAE is a dimensionality reduction-based regularized generative model that prevents overfitting. In general, a VAE is composed of an encoder-decoder block and a latent space that connects them. The input is encoded as a probability distribution throughout the latent space, and then a point is sampled from that distribution. The point is decoded, and the error can be calculated between the original data and the reconstructed data.

We evaluated the proposed approaches on 490 human metabolomics datasets retrieved from the Metabolomics Workbench [32], of which 437 studies passed quality filters (minimum 50 metabolites and 50 patients per study) and were used for evaluation. Our results show that Model-guided Agglomerative Merging performs best with Sim-R^2 of 0.72 on unseen data. Nevertheless, the Iterative Similarity-based Merging model has the best overall reconstruction score (R^2 = 0.95). Besides, when computational resources are scarce, the Iterative Similarity-based Merging provides the best performance/training time ratio with 10 times faster training time than the Model-guided Agglomerative Merging approach with a slight decrease (i.e., 5%) in imputation accuracy (with Sim-R^2 of 0.69 on unseen data).

The contributions of this paper are as follows:

1) **Two novel dataset merging strategies** designed to address the challenge of high sparsity in heterogeneous metabolomics datasets:
   - *Iterative Similarity-based Merging*, which dynamically constructs optimal merge sets based on Jaccard similarity while maintaining a sparsity threshold.
   - *Model-guided Agglomerative Merging*, which incrementally merges datasets using model-based imputation to fill non-overlapping metabolite values, forming a hierarchical ensemble of models.
2) **A unified imputation framework** that enables the training of generalizable VAE-based models across diverse datasets with non-overlapping metabolite sets—overcoming a major limitation of prior generative approaches that assume homogeneous input.
3) **A weighted Jaccard similarity metric** that incorporates metabolite value ranges to improve merge set selection and model matching for imputation.
4) **A comprehensive evaluation** on 437 real-world datasets from Metabolomics Workbench [32] (filtered from 490 retrieved studies), demonstrating superior performance (Sim-$R^2 = 0.72$) compared to state-of-the-art methods, including Gomari et al. [13].

This paper is organized as follows. A brief overview of related work is provided in Section 2, while a more comprehensive literature review, including detailed comparisons of statistical, machine learning, and generative model-based imputation methods, is available in the Supplementary Material (Section S1). Section 3 describes the proposed merging methods. In Section 4, we empirically evaluate the performance of our merging strategies and compare them to the state-of-the-art approaches. Section 5 concludes with pointers to future work.

2 Related Work

Metabolomics datasets often exhibit three types of missingness: missing not at random (MNAR), at random (MAR), and completely at random (MCAR) [6], [38]. MNAR is particularly common due to detection limits in mass spectrometry, though the true mechanism is often unknown. We designed our framework to leverage deep generative models for learning robust latent representations across heterogeneous datasets. However, we must explicitly qualify that our quantitative evaluation relied predominantly on simulated random masking. Although generative architectures are theoretically well-equipped to manage Missing Not At Random (MNAR) patterns, including LOD-style missingness, this capability is discussed here as a conceptual strength rather than a directly validated feature.

Imputation methods in metabolomics can be broadly categorized into three groups. Statistical model-based approaches include KNN, SVD, and QRILC, with QRILC performing well for MNAR and RF for MAR/MCAR [38]. Do et al. [6] evaluated 31 methods and found KNN and MICE to be effective, though MICE is computationally intensive.

Machine learning-based methods such as MAI [5] use classifiers to tailor imputation strategies based on missingness type, while NetSVR [17] incorporates network and ion interaction features. Mosaic-integration (MI) methods [11], [12], [15], [37], [42] and domain adaptation techniques [10], [25] aim to embed multi-omics data into shared latent spaces, primarily in single-cell contexts.

Generative model-based approaches, including VAEs [13] and MVAE [43], have demonstrated strong performance in imputing metabolomics data. Gomari et al. trained VAEs on homogeneous datasets, while Zhao et al. integrated genomics and metabolomics data using MVAE, though such integration is often impractical in clinical settings. Broader generative models such as GAIN [41], MIDA [14], and TabCSDI [44] have also been applied to imputation tasks in other domains.

Our work builds on these foundations by enabling robust imputation across heterogeneous metabolomics datasets without requiring prior classification of missingness or access to multi-omics data.

3 Materials and Methods

This section presents the core methodological contributions of our study, which address the challenge of imputing missing values in large-scale, heterogeneous metabolomics datasets. Unlike prior approaches that assume homogeneous metabolite coverage across datasets, our framework is designed to operate on real-world data where metabolite sets vary significantly. We introduce two novel dataset merging strategies—(i) Iterative Similarity-based Merging and (ii) Model-guided Agglomerative Merging—that enable the construction of imputation-ready datasets with reduced sparsity. These strategies are supported by a general merging framework and a weighted Jaccard similarity metric that accounts for both metabolite overlap and value range similarity. We then train Variational Autoencoder (VAE)-based models on the resulting merged datasets to perform robust, generalizable imputation. The following subsections detail the data acquisition, preprocessing, merging algorithms, and model training procedures that constitute our proposed architecture.

3.1 Database Construction and Data Preprocessing

In this study, we retrieved all human metabolomics studies (490 datasets) from the Metabolomics Workbench (as of January 2024). Section S2.1 of the Suppl. Material presents the statistics regarding the employed dataset. On the dataset, we further performed several preprocessing tasks including filtering, metabolite mapping, and normalization. The preprocessing steps are discussed in Section S2.2 of the Suppl. Material.

3.2 Merging Datasets

Given a set S of non-homogeneous metabolomics datasets, the dataset merging problem involves bringing all datasets in S to a similar feature space. Each feature in the merged dataset corresponds to a unique metabolite, and the final feature space is defined as the union of all metabolites observed across the datasets. The term "non-homogeneous" refers to the fact that datasets may have different metabolite sets. Figure 1 illustrates the non-homogeneous metabolomics dataset merging problem, where N represents the number of datasets to be merged and M_i represents the set of metabolites included in dataset S_i. Each puzzle piece denotes a metabolite in a dataset. Distinct colors represent different metabolites. After the merge operation, the union of metabolites in the datasets is obtained, and the metabolite space is unified across N datasets. Gray pieces represent the missing metabolite values to be filled with a value that will be provided by a merging algorithm.

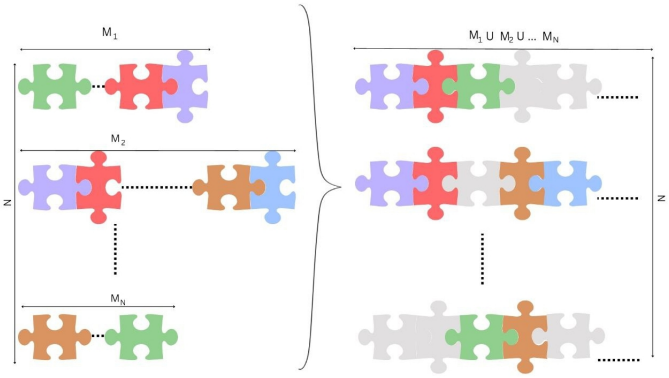

Fig. 1: Non-homogeneous Metabolomics Dataset Merging Problem. The puzzle pieces are used as a visual metaphor to represent metabolite features across datasets. Each puzzle piece corresponds to a specific metabolite, and its color indicates the identity of that metabolite. When two puzzle pieces are shown together (i.e., interlocked or adjacent), it visually represents that the corresponding metabolites are placed next to each other in a dataset. Each connected set of puzzle pieces represents a distinct dataset with the corresponding metabolite columns.

In the simplest case, consider that two datasets are being merged (Figure 2). Suppose that each dataset has two metabolites, with the green one being common to both datasets. A column representing the pink metabolite is added to dataset 1, and similarly, a column representing the orange metabolite is added to dataset 2. The newly added columns are depicted with gray colors in the merged puzzle. The pseudocode for the general merging framework is provided in the Supplementary Material, Algorithm S1.

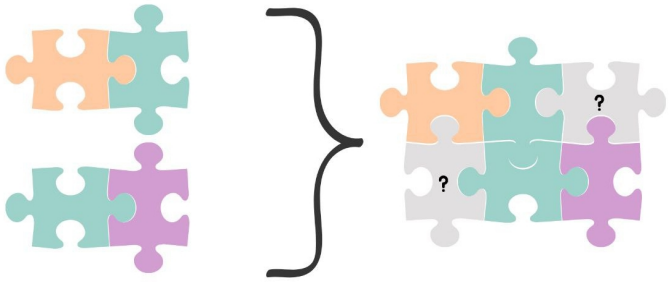

Fig. 2: Example: Merging two datasets

To illustrate the merging process, Table 1 presents a simplified example involving three metabolomics datasets (S1, S2, and S3), each containing a different subset of metabolites. The final feature space is constructed by taking the union of all metabolites observed across the datasets. Each row in the table corresponds to a dataset, and each column represents a unique metabolite. Missing values, indicated as "NA," reflect metabolites that were not originally measured

TABLE 1: Example of merging three metabolomics datasets (S1, S2, S3) into a unified feature space. Each column represents a unique metabolite, and each row corresponds to a dataset. "NA" indicates that the metabolite was not originally measured in that dataset. The final feature space is defined as the union of all metabolites across datasets.

Dataset	Met A	Met B	Met C	Met D	Met E
S1	1.2	3.4	NA	5.6	NA
S2	NA	2.1	4.3	NA	6.7
S3	0.9	NA	3.3	NA	7.1

in a given dataset. This example demonstrates how datasets with partially overlapping metabolite profiles are aligned into a unified structure suitable for downstream imputation and model training.

3.2.1 Cross-validation Framework

The merged dataset is split for training and validation of the studied dataset merging strategies. The splitting is done horizontally at the sample/patient level. Each dataset that is included in the merged data is divided into K folds. The minimum and maximum values from the training data of each fold are obtained, and then the training and validation data are normalized and yielded for training and evaluation purposes (the pseudocode is provided in the Suppl. Material, Algorithm S2)

Then, imputation models are trained based on the above-specified data splitting. Different models are trained and evaluated for each split. That is, all samples/patients from all datasets are used during the training and evaluation phases (the pseudocode is provided in the Suppl. Material, Algorithm S3).

3.2.2 Approach 1: Union-based Merging

Models trained on specific small datasets have a limited number of metabolites. In order to create more comprehensive models that include a wider range of metabolites, multiple data sets need to be combined. Union-based merging simply puts together all the individual datasets to create a single dataset (Algorithm S4 in Suppl. Material). The metabolite set of the merged dataset is simply the union of the metabolite sets in the merged datasets. The non-intersecting metabolites in each merged dataset are filled with 0s. Next, samples/patients that have less than 50 metabolites, and metabolites that occur in less than 50 samples/patients are filtered out from the merged dataset. Finally, an imputation model is trained on the merged dataset. The Metabolomics Workbench datasets exhibit a sparsity of 97.47%, covering 10563 unique metabolites. Among the metabolites, *Phenylalanine*, *Tryptophan*, and *Tyrosine* had the highest levels of completeness, with sparsity values of 24.53%, 26.76%, and 28.48%, respectively. The union-based merging represents the state-of-the-art approach, proposed by Gomari et al. [13]. We consider it as a baseline to compare our proposed approaches. While the union-based merging approach maximizes metabolite coverage, it introduces substantial sparsity due to the limited overlap among datasets. Missing values for non-intersecting metabolites are filled with default values (e.g., zeros), which can distort the data distribution and hinder model training. In contrast, our proposed merging strategies—Iterative Similarity-based Merging and Model-guided Agglomerative Merging—selectively combine datasets based on metabolite similarity and overlap. These methods incorporate sparsity control mechanisms to ensure that merged datasets remain sufficiently dense, thereby enabling the training of more robust and accurate imputation models. As a result, they outperform union-based merging in both reconstruction quality and generalization to unseen data. We next explain these novel strategies.

3.2.3 Approach 2: Iterative Similarity-based Merging

The objective of the Iterative Similarity-based Merging algorithm is to identify a subset of datasets that can be merged with the smallest sparsity possible. To ensure that merged datasets remain sufficiently dense for effective model training, the algorithm incorporates a sparsity control mechanism and a similarity-based merging strategy. For each dataset, all other datasets are ranked based on the Jaccard similarity coefficient, which quantifies the overlap between metabolite sets as the ratio of shared metabolites to the total number of unique metabolites. The merging process proceeds iteratively, starting with the most similar datasets. After each merge, the sparsity of the resulting dataset—defined as the proportion of missing values—is recalculated. If the sparsity exceeds a predefined threshold (denoted as θ), the merging process is halted. If the threshold is exceeded after the first merge attempt, the original dataset is retained as its own merge set. The sparsity threshold θ thus plays a critical role in balancing dataset size and data completeness. A lower θ results in smaller but denser merge sets, which are more suitable for accurate model training, while a higher θ allows for larger merge sets at the risk of introducing more missingness. This combined use of Jaccard similarity and sparsity control ensures that only datasets with sufficient overlap are merged, thereby maintaining a minimum level of completeness and avoiding the creation of overly sparse merged datasets that would hinder model performance. In particular, for each dataset, an optimal merge set is constructed and recorded (Figure 3). Algorithm 1 provides a pseudo-code for Iterative Similarity-based Merging. First, pairwise Jaccard similarities (Eq. 1) are calculated between each dataset pair based on shared metabolites (lines 2-5). Then, for each dataset, the remaining datasets are ranked based on their similarity (line 7). Next, the merge operation is performed in the order of dataset similarity scores (lines 12-23). After each iteration, the sparsity of the merged dataset is calculated (line 15). The sparsity of a dataset is the ratio of empty cells in the dataset. If the sparsity of a merged set exceeds a predefined threshold, the merging stops. In case the sparsity exceeds the threshold after the first merge attempt, the initial dataset is returned as the merge set of itself (i.e., identity merge set) (lines 16-20). Due to the possibility of completely overlapping merge sets, identical merge sets are removed (line 24). For each dataset in the database, a model is trained on its merge set to impute new unseen data (lines 25-27). We give an example.

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

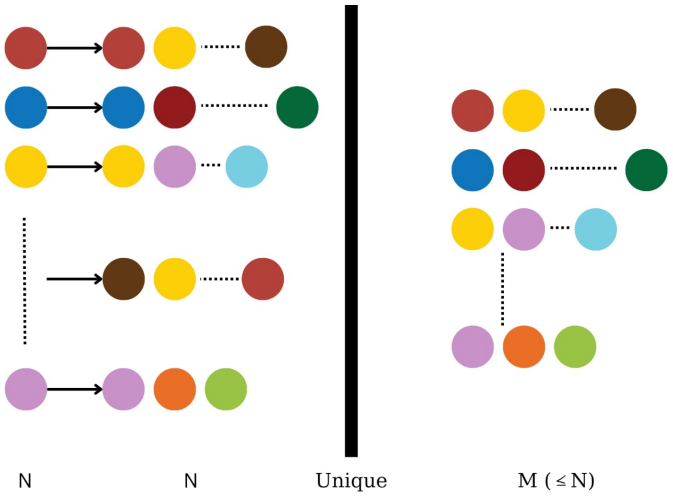

Fig. 3: Illustration of the dataset merging phase of *Iterative Similarity-based Merging*. Colors represent different datasets. An optimal merge set for each dataset is created (i.e., N merge sets for N datasets). Duplicate merge sets are filtered.

Example: Assume that the repository of datasets includes S_1, S_2, S_3, and S_4 with metabolites provided in Figure 4a, and the sparsity threshold θ is 0.3. Pairwise Jaccard similarities of datasets based on their metabolite sets are shown in a matrix (Figure 4b). The iteration starts with S_1. S_3 is the most similar dataset. S_1 and S_3 are merged. The sparsity of the merged set is $0.2 < \theta$. Hence, the merging process continues. The second most similar dataset to S_1 is S_2. Next, S_2 is merged, and the new sparsity is calculated as $0.35 > \theta$. Therefore, the merging iterations end for S_1. The optimal merge set for dataset S_1 is $[S_1, S_2]$. The process is repeated for each of the remaining datasets, and optimal merge sets are created (Figure 4c). Since the merge sets computed for S_2 and S_4 are the same, one of them is eliminated. Then, three models are trained based on distinct merge sets at the end of the process

.

When new, unseen data arrives, models are scored and sorted based on the similarity between the new data and the merge set for each model using Jaccard scores based on metabolite sets. The new data is imputed using a predefined number of most similar models, and the final value is computed as the weighted average of the predicted values by each model based on the similarity scores of the models (Eq. 2).

$$y_{ij}^{final} = \frac{\sum(y_{ij}^{pred} * sim\text{-}score)}{\sum sim\text{-}score} \tag{2}$$

The original Jaccard score does not consider the similarity of ranges for overlapping metabolite pairs. In order to enhance the process of creating merge sets, we propose a weighted Jaccard score. In particular, each common metabolite between two datasets contributes to the intersection in the numerator of Jaccard by the similarity of their ranges (Eq. 3 where $mets(S_i)$ represents metabolites of dataset S_i, min_{S_k} and max_{S_k} represent the minimum and maximum values of metabolite m_i in dataset S_k, respectively).

Algorithm 1: Iterative Similarity-based Merging

Input : Metabolite datasets in database: $datasetList$, Sparsity threshold: θ
Output: Similarity based models: M_{iter_N}

Initialize: $mergedDatasetDB$;
foreach *pair of datasets* (S_1, S_2) *in* $datasetList$ **do**
  $m_1 \leftarrow S_1.metabolite\_names$;
  $m_2 \leftarrow S_2.metabolite\_names$;
  Calculate *jaccard_similarity*(m_1,m_2);
foreach *dataset* S *in* $datasetList$ **do**
  $rankedList \leftarrow$ **Rank** datasets with Jaccard sim ;
  $initialDataset \leftarrow$ `S.data`;
  $mergedDataset \leftarrow$ `S.data`;
  $previousMergedDataset \leftarrow$ `None`;
  $sparsity \leftarrow$ *calculate_sparsity*($mergedDataset$);
  foreach *dataset* S_{MS} *in* $rankedList$ **do**
    $vals \leftarrow$ initialize an array of 0s;
    $mergedDataset \leftarrow MergeDatasets([S, S_{MS}], vals)$;
    $sparsity \leftarrow$ *calculate_sparsity(merged dataset)* ;
    if $sparsity > \theta$ **then**
      if $previousMergedDataset$ *is not* None **then**
        yield $previousMergedDataset$
      else
        yield $initialDataset$
    $previousMergedDataset \leftarrow mergedDataset$;
  yield $mergedDataset$
  Add: yielded dataset to $mergedDatasetDB$;
Remove: duplicate entries in $mergedDatasetDB$;
foreach $mergedDataset$ *in* $mergedDatasetDB$ **do**
  $M_{iter_N} \leftarrow GenerateVAEModel(mergedDataset)$;
  Store: M_{iter_N} to be used for imputing;

$$WJ(S_1, S_2) = \frac{\sum_{m_i \in mets(S_1) \cap mets(S_2)} (1 - S_{m_i})}{|mets(S_1) \cup mets(S_2)|}$$
$$S_{m_i} = \frac{|min_{S_1} - min_{S_2}| + |max_{S_1} - max_{S_2}|}{max(max_{S_1}, max_{S_2}) - min(min_{S_1}, min_{S_2})} \tag{3}$$

3.2.4 Approach 3: Model-guided Agglomerative Merging

The integration of diverse datasets significantly increases the number of models to be utilized. However, it introduces the challenge of sparsity. To address the sparsity problem, we propose another method to fill the non-intersecting metabolites in a dataset by treating them as observed values based on the predictions of cross-dataset models.

$$S_1 \leftarrow [M_1, M_2, M_3]$$
$$S_2 \leftarrow [M_1, M_4, M_5]$$
$$S_3 \leftarrow [M_2, M_5]$$
$$S_4 \leftarrow [M_1, M_5, M_6, M_7]$$

(a) Sample datasets and their metabolites

	S_1	S_2	S_3	S_4
S_1	1	0.2	0.25	0.17
S_2	0.2	1	0.25	0.4
S_3	0.25	0.25	1	0.2
S_4	0.17	0.4	0.2	1

(b) Jaccard similarity matrix of metabolite names

$$S_1 \leftarrow [S_1, S_3]$$
$$S_2 \leftarrow [S_2, S_4, S_3]$$
$$S_3 \leftarrow [S_3, S_1, S_2]$$
$$S_4 \leftarrow [S_4, S_2, S_3]$$

(c) Optimal merge sets for each dataset

Fig. 4: Example of Iterative Similarity-based Merging

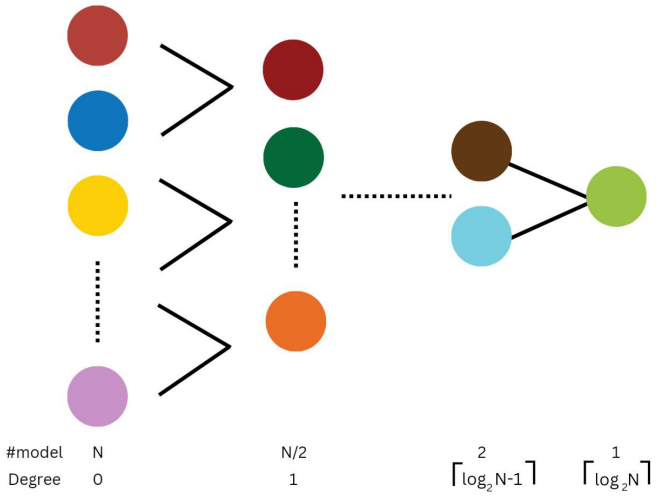

Fig. 5: The circles represent different models, while the colors represent the datasets on which models are trained. At degree 0, N models are created for each of the N datasets. Since the merging operation continues until a single model containing all metabolites and a single dataset is created, there is one model formed at the final step. As the number of nodes is halved at each step, the maximum degree becomes $\lceil \log 2N \rceil$.

In this approach, datasets are combined in pairs at each step. The merging process continues until all datasets are used in a single model. The merging order of datasets is represented by a binary tree where each node represents an imputation model that is trained on an individual dataset (at the leaf level) or a merged dataset (at internal levels). "Degree" of a node refers to the height of the node in the merging tree data structure, where the leaf level has a height value of 0, while the root has a height value of $log_2 N$ when N datasets are merged. (see Figure 5).

Note on Visual Representation: To accurately convey the architecture of the Model-guided Agglomerative Merging framework, the following schematics (Figures 5–7) shift from the data-centric visualizations used in previous sections to an abstract, process-oriented visual language. In these diagrams, circles represent trained generative model nodes within the hierarchical tree (rather than raw datasets), while the accompanying pathways illustrate the algorithmic flow of training, evaluating, and merging these latent spaces at progressively higher degrees.

Algorithm 2 outlines the steps carried out in this approach. At degree d, a model is created for each of the datasets available at this degree level, and stored at the corresponding degree (lines 4 - 6). Then, for each unique pair of datasets, Jaccard similarity is computed based on their metabolite sets (lines 8 - 12). Next, starting from the most similar pair, datasets are merged in pairs in the current degree level (lines 14 - 24). The models of datasets $S1$ and $S2$ are used reciprocally to fill the non-overlapping metabolites in the other data set before merging (lines 19 -21). Figure 6 illustrates this step. The merged datasets are added to the new pool, which will be used in the next step (line 22). $S1$ and $S2$ datasets are then removed from the current pool as well as the similarity matrix. This process continues until no datasets remain to be merged (line 3). If there is only one dataset left in the pool (happens when the number of datasets at the current degree level is odd), it is added to the new pool to be used in the next step (line 26). Afterwards, the current dataset pool pointer is switched to the next degree's dataset pool, and the degree is increased by one (lines 27 - 28). Finally, all the obtained models are saved with their degree information. As shown in Figure 5, the maximum degree will be $\lceil \log_2 N \rceil$, resulting in $2N - 1$ unique models.

Once the models at each degree level are created during training, the model-guided agglomerative merging approach is ready to be used to impute missing values in a new (unseen/future) dataset. Algorithm 3 outlines the imputation procedure. To impute a dataset S_new, at each degree level, the dataset S_M that is most similar to S_{new} is identified (lines 4 - 6). Then, the model that was created on S_M during the training is used to impute S_{new} (line 8). Then, the imputation values that are provided by the most similar dataset model at each degree level are combined by taking their weighted average (lines 10 - 11).In particular, a $scale$ parameter is used to adjust the contribution level of higher-degree models. As the value increases, the impact of higher-degree models on the final imputation value decreases. Given the significant computational requirements of the hierarchical merging process, we utilized a scale factor of 1 based on empirical testing on a subset of the datasets. This configuration was chosen to effectively stabilize the imputation using consensus from higher-degree models while maintaining computational efficiency and avoiding the overhead of exhaustive per-fold hyperparameter optimization. Next, we give an example to

Algorithm 2: Model-guided Agglomerative Merging-based Model Generation

Input : List of datasets: $DatasetList$
Output: Saved imputation models

$degree \leftarrow 0$;
$vaeModels \leftarrow$ define array;
while $DatasetList$ *has element* **do**
  foreach *dataset* S *in* $DatasetList$ **do**
    $model \leftarrow GenerateVAEModel(dataset)$;
    $vaeModels[degree][S] \leftarrow model$;
  $sims \leftarrow$ initialize an empty list;
  foreach *unique pair of datasets* (S_1, S_2) *in* $DatasetList$ **do**
    $m_1 \leftarrow S_1.metabolite\_names$;
    $m_2 \leftarrow S_2.metabolite\_names$;
    $similarity \leftarrow jaccard\_similarity(m_1, m_2)$;
    $sims[S_1, S_2] \leftarrow similarity$;
  $nextDatasetList \leftarrow$ initialize an empty list;
  while $sims$ *has 2 or more elements* **do**
    $vals \leftarrow$ define array;
    $S_1, S_2 \leftarrow$ Most similar pair in $sims$;
    $M_1 \leftarrow vaeModels[degree][S_1]$;
    $M_2 \leftarrow vaeModels[degree][S_2]$;
    $vals[S_1] \leftarrow M_2(S_1)$;
    $vals[S_2] \leftarrow M_1(S_2)$;
    $S_{1,2} \leftarrow MergeDatasets(list(S_1, S_2), vals)$;
    $nextDatasetList.append(S_{1,2})$;
    Remove: S_1, S_2 from $DatasetList$;
    Remove: all pairs of S_1, S_2 from $sims$;
  if $DatasetList$ *has a single element* S **then**
    $nextDatasetList.append(S)$;
  $DatasetList \leftarrow nextDatasetList$;
  $degree \leftarrow degree + 1$;
Store: $vaeModels$ for all degree and all datasets

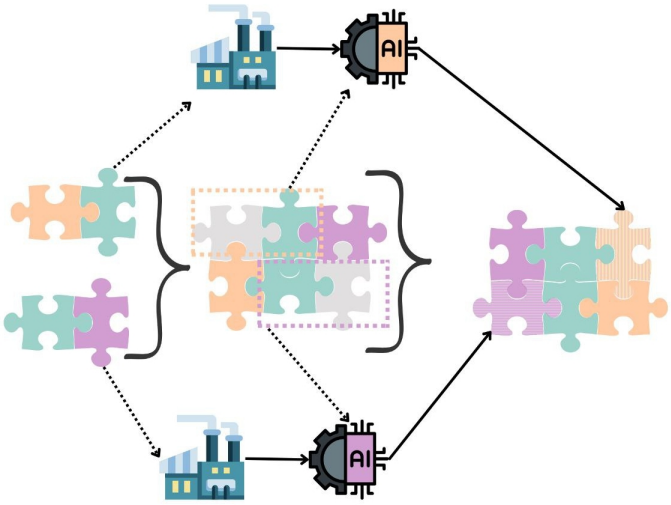

Fig. 6: This figure illustrates the operations performed at each merging step in Figure 5. The colors represent the metabolites, while the factory and gear represent the imputation model stages created from the relevant datasets. The green piece represents the common metabolites in the two datasets. The orange and purple pieces need to be filled with the corresponding models. The model associated with the dataset S_{gp} of green and purple pieces fills the purple area in the other dataset S_{go} with green and orange pieces. On the other hand, the orange area in S_{gp} is similarly filled with the model trained on S_{go}. Then, the two datasets are merged. The similarity in color but jagged edges of the model outputs emphasizes the dependency on the model.

illustrate how Model-guided Agglomerative Merging-based model generation and imputation work.

Example: Consider that we have 4 different datasets (see Figure 7): S_1, S_2, S_3, and S_4. S_1 has metabolites M_1, M_2, and M_3; S_2 has metabolites M_1, M_4, and M_5; S_3 has metabolites M_2 and M_5; and S_4 has metabolites M_1, M_5, M_6, and M_7. Initially, imputation models are created for S_1, S_2, S_3, and S_4 at degree 0. Let these models be IM_1, IM_2, IM_3, and IM_4, respectively. Next, a similarity matrix is created between the datasets. S_2 and S_4 are the most similar datasets with common metabolites M_1 and M_5. $S_4 \setminus S_2$ (i.e., the difference of S_4 from S_2) contains metabolites M_6 and M_7. To fill the unmatched metabolites in S_2 with IM_4, the IM_4 model is provided as input with M_1 and M_5 with original values in S_2, while M_6 and M_7 are left empty to be imputed by IM_4. Likewise, $S_2 \setminus S_4$ contains metabolite M_4. The IM_2 model is supplied with M_1 and M_5 with the original values in S_4, while M_4 is left empty to be imputed by IM_2. This process is also illustrated in Figure 6. Then, the merged dataset S_{24} is obtained, and it is stored for the next step with the newly imputed values. S_2 and S_4 are removed from the dataset pool and the similarity matrix. The same process is repeated for S_1 and S_3. At this point, since there are no elements left in the similarity matrix, the dataset pool is populated with S_{24} and S_{13}, and the degree is increased to 1 for the next iteration. Now, S_{24} has metabolites M_1, M_4, M_5, M_6, and M_7, and S_{13} has metabolites M_1, M_2, M_3, and M_5 for further processing. The model creation and similarity calculation processes are repeated. The similarity between the two datasets is 0.29, and they are merged similarly. The dataset S_{1234} is created and placed in the dataset pool for the next iteration. Since there are no datasets to merge, the degree is increased to 2, and the dataset pool is updated to be empty. The process ends, and the models for degrees 0, 1, and 2 are saved.

When imputing a new future dataset S' with metabolites M_1, M_3, M_5, M_7, and M_8, first, similarity values between S' and all datasets (i.e., S_1, S_2, S_3, S_4) at the leaf level (i.e., degree 0) are computed as 0.33, 0.33, 0.17, and 0.50, respectively. Since the most similar dataset is S_4, its corresponding model, $IM4$, is selected for imputation at degree 0. More specifically, $IM4$ model is provided with an input dataset that contains metabolites M_1, M_5, M_6, and M_7. That is, the original values of metabolites M_1, M_5, and M_7 from S' are kept, while leaving M_6 as empty to be imputed by $IM4$. The model's output serves as an imputation value at degree 0 for the empty field (i.e., M_6) in S'. At the next degree level (i.e., degree 1), with similarities of 0.43 for S_{24} and 0.50 for S_{13}, the IM_{13} model is used for imputation. The model is

Algorithm 3: Model-guided Agglomerative Merging-based Imputation

Input : S_{new}: A new dataset that will be imputed
Output: One fold of the merged dataset, where the data has been normalized

$vaeModels \leftarrow$ **Load** imputation models;
$maxDegree \leftarrow$ **Length** of $vaeModels$;
$imputationValues \leftarrow 0$;
for $degree \leftarrow 0$ **to** $maxDegree$ **do**
  $datasetModels \leftarrow vaeModels[degree]$;
  $S_{MS} \leftarrow$ **Find** most similar dataset to S_{new} in $datasetModels$;
  $model \leftarrow vaeModels[degree][S_{MS}]$;
  $values \leftarrow model(S_{new})$;
  $values \leftarrow$ **Get** imputation values from empty cells in $values$;
  $imputationValues \leftarrow imputationValues + values * scale^{-degree}$;
$imputationValues \leftarrow imputationValues / \sum_{degree=0}^{maxDegree} scale^{-degree}$;
return $imputationValues$

provided with a dataset that contains M_1, M_2, M_3, and M_5 fields. M_1, M_3, and M_5 are filled with the existing values from S′, while M_2 is left empty to be imputed by IM_{13}. At degree 2, there is only one model (i.e., IM_{1234}). It is used to obtain imputation values for metabolites M_2 and M_4. Finally, the imputed fields from three degrees as weighted-scaled by the degree.

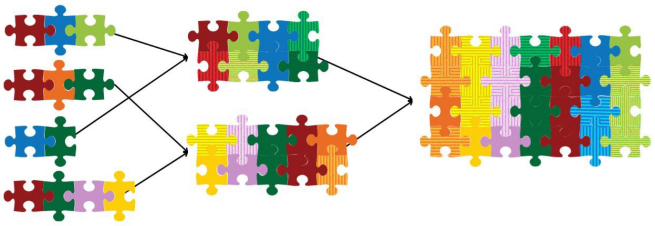

Fig. 7: Merging four different datasets with the agglomerative model-guided approach. The colors represent unique metabolites. The jagged colors represent the values of the metabolites that are filled with corresponding models.

3.3 Illustrative Example

To enhance the interpretability of our proposed merging and imputation strategies, we present a simplified, illustrative example using synthetic metabolomics datasets. This example reflects common challenges in real-world metabolomics studies, where datasets often contain partially overlapping metabolite profiles and varying degrees of missingness.

Table 2 shows the initial synthetic datasets (S_1, S_2, and S_3) with measurements for a subset of four metabolites (M_1–M_4). Several values are missing, simulating the sparsity typically observed in public metabolomics repositories.

TABLE 2: Initial synthetic metabolomics datasets before merging. Missing values are indicated with em dashes.

Dataset	M1	M2	M3	M4
S_1	1.20	0.80	—	—
S_2	—	0.90	1.50	—
S_3	1.10	—	—	0.70

We then apply three different merging strategies—Union-based, Iterative Similarity-based, and Model-guided Agglomerative Merging—to these datasets. Table 3 summarizes the resulting merged datasets and highlights the imputed values in red. Below, we walk through how each merging strategy processes these datasets. This step-by-step example provides an intuitive understanding of how each strategy handles missing data and constructs a unified dataset for downstream imputation.

1. Union-based Merging

Step 1: Take the union of all metabolites: {M1, M2, M3, M4}
Step 2: Align all datasets to this unified metabolite space
Step 3: Fill missing values with zeros

2. Iterative Similarity-based Merging

Step 1: Compute Jaccard similarity between datasets based on shared metabolites:

- S_1 and S_2 share M2 ⇒ similarity = 1/3
- S_1 and S_3 share M1 ⇒ similarity = 1/3
- S_2 and S_3 share no metabolites ⇒ similarity = 0
- Assume that the sparsity threshold θ is 0.3

Step 2: Form merge-sets:

$$S_1 \leftarrow [S_1, S_2, S_3]$$
$$S_2 \leftarrow [S_2, S_1]$$
$$S_3 \leftarrow [S_3, S_1]$$

Step 3: Suppose that column-wise mean is used as a simple imputation model. Fill missing values using column-wise mean in each merge-set.

3. Model-guided Agglomerative Merging

Step 1: Train initial models on individual datasets
Step 2: Merge datasets in pairs based on similarity
Step 3: Use trained models to impute missing values in each other
Step 4: Continue merging until a single dataset remains

3.4 Imputation Model

The variational autoencoder (VAE) [20] is an extension of autoencoders designed for generative image sampling. In a variational autoencoder, the initial data is sampled from a parameterized prior distribution $p_\theta(\mathbf{z})$ instead of a fixed compressed space. Encoder-decoder blocks are trained together so that the model minimizes a joint loss, reconstruction error $\mathcal{L}_{MSE}$, and Kullback-Leibler divergence between the parametric posterior $q_\phi(\mathbf{z}|\mathbf{x})$ and the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$. The loss function is expressed as in Eq. 4.

TABLE 3: Illustrative example of merging strategies with imputed values highlighted in red. Three small datasets (S_1, S_2, S_3) with partially overlapping metabolites (M_1–M_4) are shown. **Top:** Union-based merging fills missing values with zeros, resulting in a sparse matrix. **Center:** Iterative Similarity-based Merging merges S_1 and S_2 based on Jaccard similarity and imputes missing values using column-wise means; S_3 is treated separately. **Bottom:** Model-guided Agglomerative Merging fills missing values using row-wise means (simulating model-based predictions), producing a more balanced matrix.

Dataset	M1	M2	M3	M4
Union-based Merging				
S_1	1.20	0.80	0	0
S_2	0	0.90	1.50	0
S_3	1.10	0	0	0.70
Iterative Similarity-based Merging				
S_1	1.20	0.80	1.50	0.70
S_2	1.20	0.90	1.50	–
S_3	1.10	0.80	–	0.70
Model-guided Agglomerative Merging				
S_1	1.20	0.80	1.40	0.60
S_2	1.15	0.90	1.50	0.65
S_3	1.10	0.85	1.45	0.70

$$\begin{aligned} \mathcal{L}_{VAE} &= \mathcal{L}_{MSE} + \mathcal{L}_{KL} \\ \mathcal{L}_{VAE}(\theta,\phi) &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2 \\ &+ \quad D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})) \end{aligned} \tag{4}$$

To handle the varying number of metabolites across different merge sets, we implemented a dynamic architecture initialization. While the depth of the VAE and the expansion/reduction factors for the hidden layers remained consistent across all models, the input and output layer dimensions were configured dynamically to match the exact number of unique metabolites present in each specific merged dataset. This ensured that the model capacity scaled appropriately with the data complexity of each merge set.

As the VAE encoder requires a fully populated input vector, all datasets undergo an initial imputation step using kNN prior to passing through the VAE. This provides the VAE with a complete, albeit noisy, input matrix, which the model subsequently refines through its probabilistic latent space representation.

Our baseline *MetaboliteVAE* model architecture includes multiple fully connected linear layers. Once the initial missing values are pre-filled using KNN imputation, they are passed to the encoder layer. The encoder is then followed by the *LeakyRELU* activation function with a negative slope of 0.2, which is formulated as $z = max(0, a) + 0.2 * min(0, a)$. The size of the encoder layers is reduced by dividing by two, while the decoder layers are expanded by multiplying by two. The latent dimension size is tuned within the given bounds to minimize training loss. A hyperparameter called kld-weight is set to 0.01 to adjust the contribution of $\mathcal{L}_{KL}$ term in the loss function. Adam optimizer is used with a learning rate of 0.001. The model architectures are implemented using PyTorch 1.13.1 [27] with CUDA version 11.7 on Python 3.8.12. Tensor operations are performed on both the CPU and GPU.

A re-parameterization trick is applied while sampling the latent vector to make the stochastic sampling process deterministic and backpropagate the gradient, making the model trainable. The most common choice for parametric posteriors is a multivariate Gaussian distribution. In that case, VAE learns from the mean and variance of the distribution and a random noise ϵ sampled from the Gaussian. The re-parameterization trick is expressed in Eq. 5.

Using VAE architectures with large-scale metabolomics data is promising to learn more accurate latent dimension representations that systematically account for non-linear effects. In addition, the probabilistic structure of VAEs could learn latent dimensions that are generalizable across multiple datasets [5].

$$\begin{aligned} &\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{2(i)}\boldsymbol{I}) \\ &\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}) \end{aligned} \tag{5}$$

In our proposed framework, the Variational Autoencoder (VAE) serves as the core generative model for imputing missing metabolite values. After datasets are merged using one of the proposed strategies, the VAE is trained to learn a compact latent representation that captures the underlying structure and correlations among metabolites. This latent space enables the model to reconstruct missing values in a principled and data-driven manner. To prevent overfitting—especially important in high-dimensional and sparse metabolomics data—the VAE incorporates a regularization term in its loss function: the Kullback-Leibler (KL) divergence between the learned posterior distribution and a standard normal prior. This regularization encourages the latent space to remain smooth and continuous, promoting generalization across diverse datasets. Additionally, the stochastic sampling of latent variables during training (via the reparameterization trick) introduces controlled noise, which acts as implicit regularization and further enhances robustness.

Compared to traditional dimensionality reduction techniques such as Principal Component Analysis (PCA) and Kernel PCA (KPCA), Variational Autoencoders (VAEs) offer several advantages for imputing missing values in metabolomics data. PCA is limited to capturing linear relationships, while KPCA extends this to non-linear mappings but requires careful kernel selection and is computationally intensive. In contrast, VAEs learn a non-linear, probabilistic latent representation of the data through deep neural networks, enabling them to model complex dependencies among metabolites. Additionally, the KL divergence term in the VAE loss function acts as a regularizer, promoting smoothness in the latent space and reducing overfitting. This is particularly beneficial in metabolomics, where datasets are often sparse and noisy. These properties make VAEs more robust and effective for reconstructing missing values across diverse and heterogeneous datasets.

The personal computer was equipped with a GTX 1080Ti graphics card, which had 11 GB of video random access memory (VRAM), an Intel(R) Core(TM) i7-8700K central

processing unit (CPU) running at a speed of 5.0 gigahertz, along with 16 gigabytes of random access memory (RAM).

4 Experimental Evaluation

4.1 Metrics

As a performance metric, R^2 (R-squared) score, also known as the coefficient of determination, is utilized to calculate the robustness of simulated missingness (Eq. 6). It is originally a regression metric, which serves as a statistical indicator of the degree to which the regression line accurately represents the observed data. The range of values that R^2 can take is in the interval $[-\infty, 1]$. A value of $R^2 = 1.0$ indicates a perfect fit, while a value of $R^2 < 0$ indicates that the model cannot meaningfully represent the given data. To evaluate the performance of the models based on only the imputed values across datasets, the $Sim - R^2$ score is calculated based on the masked values applied prior to the cross-validation operation (Eq. 7).

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum(y_i - y_{pred})^2}{\sum(y_i - y_{avg})^2} \tag{6}$$

$$\begin{aligned} Sim - R^2 &= 1 - \frac{SS_{res(mask)}}{SS_{tot(mask)}} \\ &= 1 - \frac{\sum(y_{i(mask)} - y_{pred(mask)})^2}{\sum(y_{i(mask)} - y_{avg(mask)})^2} \end{aligned} \tag{7}$$

To ensure statistical robustness and account for variability due to random data splits, we repeated the 10-fold cross-validation procedure 10 times using different random seeds. For each method and configuration, the average Sim-R² and R² scores across these 10 repetitions were reported as the final evaluation metrics. This approach provides a more reliable estimate of model performance and reduces the influence of outlier folds or seed-specific effects.

Furthermore, while $Sim - R^2$ provides a global measure of imputation accuracy, relying solely on it for log_{10} transformed data can be overly optimistic due to scale compression. To provide a stringent, scale-independent evaluation of biological concordance, we also report the median within-dataset Spearman correlation coefficient (ρ). This metric evaluates how well the imputed values preserve the relative rank-order abundance of each metabolite within its specific experimental context. For each metabolite j, Spearman correlation ρ_j is computed between imputed and pseudo-ground-truth values across all simulated-missing positions in the test set:

$$\rho_j = \text{Spearman}\big(\hat{y}_j^m,\, y_j^m\big) \tag{8}$$

where $\hat{y}_j^m$ and y_j^m denote the imputed and pseudo-ground-truth values at simulated-missing positions for metabolite j, respectively. Metabolites with fewer than three valid observations are excluded. We report the median and mean Sim-Spearman across all valid metabolites (Sim-Spearman-Median, Sim-Spearman-Mean). A high Spearman score confirms that the model preserves rank-order concordance independently of absolute scale.

4.2 Experimental Setup

To rigorously evaluate imputation accuracy, we simulated missingness on the strictly held-out test splits of our 10-fold cross-validation framework. It is important to explicitly note that because true instrument-measured values are unavailable for the naturally missing entries in these sparse datasets, we established a pseudo-ground truth. This was achieved by first fully imputing the unseen test split using KNN, and subsequently masking 10–20% of these values. The $Sim - R^2$ metric is therefore calculated against this pseudo-ground truth baseline. Furthermore, to ensure the integrity of the unseen data evaluation, these test splits were completely isolated from all prior dataset merging and VAE training phases.

4.3 Comparison of Different Initial Imputation Methods

Our database includes multiple metabolomics datasets with various missingness ratios. In our study, initial missing value imputation serves two distinct purposes. First, for evaluation purposes, we fully imputed these datasets to establish a complete "ground truth". This allowed us to artificially mask known values and rigorously evaluate our models' reconstruction performance on novel data. Second, in the practical application of our framework, generative models like VAEs require complete input matrices (i.e., no unhandled NaNs) to pass data through the encoder. Therefore, an initial pre-imputation step is strictly required for any new, unseen dataset prior to VAE processing. Based on our evaluation of different alternatives (Section S3.1 in Suppl. Material), kNN imputation is chosen as the standard initial missing value filling method for our proposed merging approaches, as it provides a stable baseline initialization that the VAE subsequently refines.

4.4 Comparison of Different Normalization Schemes

In order for the *MetaboliteVAE* model to learn effectively, the data needs to be normalized, scaled, or transformed after the imputation of initial missing values. Otherwise, the larger raw values in a dataset could hinder the model's ability to backpropagate and optimize the weights and biases of the layers present in the encoder and decoder parts of the VAE. To this end, we evaluate twelve different data pre-treatment methods (Section S3.3 in Suppl. Material), and the log_{10} transformation was chosen as the standard pre-treatment method for our proposed merging approaches.

4.5 Comparison of Dataset Merging Approaches

We next evaluate our proposed dataset merging strategies for missing metabolite value imputation. The first approach, *Union-based Merging*, aims to join all available datasets in public databases to create one universal large model. This approach represents the state-of-the-art in the metabolomics data imputation space [13]. However, under the *Union-based Merging* scheme, *MetaboliteVAE* fails to learn and optimize parameters. To form a comparable baseline, we attempted to make union-based merging work by introducing an improved version of it. We call the improved version *Best Effort Union-based Merging*. It selects random seed datasets from our database as a starting point and joins them based

on the Jaccard similarity score until the merged dataset does not lead to a trainable model (i.e., the loss value is not undefined or NaN). The number of most similar datasets that are merged in this approach is a parameter, represented as n_{sim}. Different n_{sim} values (e.g., 16, 12, 10, 8) are evaluated, and we determine that 8 is the most optimal number to create trainable models under the best-effort union-based merging approach before trained models lead to undefined loss values. 10 seed datasets were randomly chosen from the Metabolomics Workbench database, and each of them was combined with their 8 most similar datasets. A model is trained for each merged dataset using a 10-fold cross-validation setup. In cross-validation, datasets are first shuffled, then divided into 90% training and 10% test splits, repeated by the number of folds. Trained models were saved to be used in prediction. For 437 studies in the Metabolomics Workbench database, the test data that was reserved in the cross-validation split was used as unseen data. These test dataset splits are fully imputed with the *KNNImputer*. Then, artificial missing values are created at random missingness rates between 10%-20%, and predictions are obtained using different numbers of the most similar models (parametrized as n_{best}). We experiment with different n_{best} values of 1, 5, and 10. Our evaluation revealed that setting $n_{best} = 1$ yielded the highest imputation accuracy for the Iterative Similarity-based Merging approach. This indicates that averaging predictions across multiple distinct models ($n_{best} > 1$) during inference introduces biological noise from less-similar latent spaces, which dilutes the high-confidence prediction of the single best-matched model. It is important to emphasize that this optimal $n_{best} = 1$ configuration does not negate the integrative nature of the approach; the single selected model is itself the product of an optimized, multi-dataset merge set constructed during the training phase. Consequently, this approach produces standalone VAE models that are highly robust without requiring inference-time model ensembling. The results are shown in Figures S2 and S3 (Suppl. Material). For cases where we have predictions by multiple models (i.e., $n_{best} > 1$), Jaccard similarity-based weighted averages were used.

To compare performance across the repository, we evaluated our proposed models on the full set of 437 datasets. While the union-based baseline approach sought to maximize metabolite coverage, it was constrained to a subset of 63 datasets due to training instabilities (i.e., NaN loss values) that occurred when attempting to merge higher volumes of sparse, heterogeneous data. While we acknowledge that a direct head-to-head comparison on a common intersection of datasets would be ideal, we believe the evaluation on the full range of data each model is capable of processing provides the most accurate reflection of its functional scalability and real-world robustness.

For the second approach, *Iterative Similarity-based Merging*, for each of 437 studies in our Metabolomics Workbench database, an optimal merge set is created with the proposed algorithm. After duplicates are eliminated, we obtain 264 unique Jaccard-based merge sets. For each merge set, studies inside the set are merged, and a model is trained using a 10-fold cross-validation setup on the merged dataset as explained above. After saving our models in a repository, we tested them with the test data from each dataset that we split with cross-validation. Finally, the artificially created missing values are imputed with our newly proposed *IterativeSimilarityImputer*. The test data of 437 studies are imputed with different values of n_{best} as 1, 5, and 10. The results are presented in Figures S4 and S5 (Suppl. Material).

The third dataset merging approach, *Model-guided Agglomerative Merging*, merges datasets two by two until a single model is produced. When evaluating the training losses during the hierarchical joining process, we observed that the loss value of the joint model becomes NaN (Not-a-Number) after successfully joining 64 models. This instability arises from compounding sparsity, similar to the failure mode observed in our baseline *Union-based Merging*. Consequently, rather than forcing the agglomeration all the way to a single universal root node, the joining process is explicitly early-stopped just before the loss values become undefined. In practice, this means training begins with 128 individual dataset models at degree 0 and merges them pairwise; the process is halted after 64 merge operations (i.e., at degree 6 of the binary tree). This early-stopped ensemble is the configuration explicitly referred to as *with 128 joint* in our comparative results. The performance of this model-guided agglomerative merging on unseen Metabolomics Workbench data is presented in Figure S6 (Suppl. Material). Finally, to compare different approaches within the same chart, we present the performance of the best configuration for each studied approach in Figure 8. Among all the presented approaches, model-guided agglomerative merging performs the best with an average Sim-R^2 of 0.72. Furthermore, when evaluating within-dataset metabolite concordance, the Model-guided Agglomerative Merging approach maintained the best median Spearman correlation (i.e., 0.50), indicating that the framework better preserves local biological variation in addition to global numerical accuracy (Table 4).

To further investigate the variance in imputation performance, we analyzed the characteristics of studies in the Metabolomics Workbench repository against their $Sim-R^2$ scores using the Model-guided Agglomerative Merging approach (Fig. 9). The resulting distribution is bimodal, revealing two distinct performance clusters: a high-performing group ($Sim - R^2 \geq 0.3$, $n = 395$) and a lower-performing group ($Sim - R^2 < 0.3$, $n = 61$). Comparative analysis shows that the lower-performing group is characterized by significantly higher metabolite counts (median 398 vs. 157, $p < 0.001$) and elevated initial sparsity levels. This analysis provides a practical diagnostic for researchers: datasets with high feature dimensionality and extreme sparsity represent more complex imputation challenges for current generative models, potentially requiring further refinement of hierarchical integration strategies for such cases.

4.6 Comparison to the State-of-the-Art

To contextualize the performance of our framework, it is crucial to distinguish between local and global imputation paradigms. Traditional statistical methods (e.g., KNN, MICE) and machine learning algorithms (e.g., Random Forest) represent the state-of-the-art for local imputation within single, relatively dense datasets. However, these

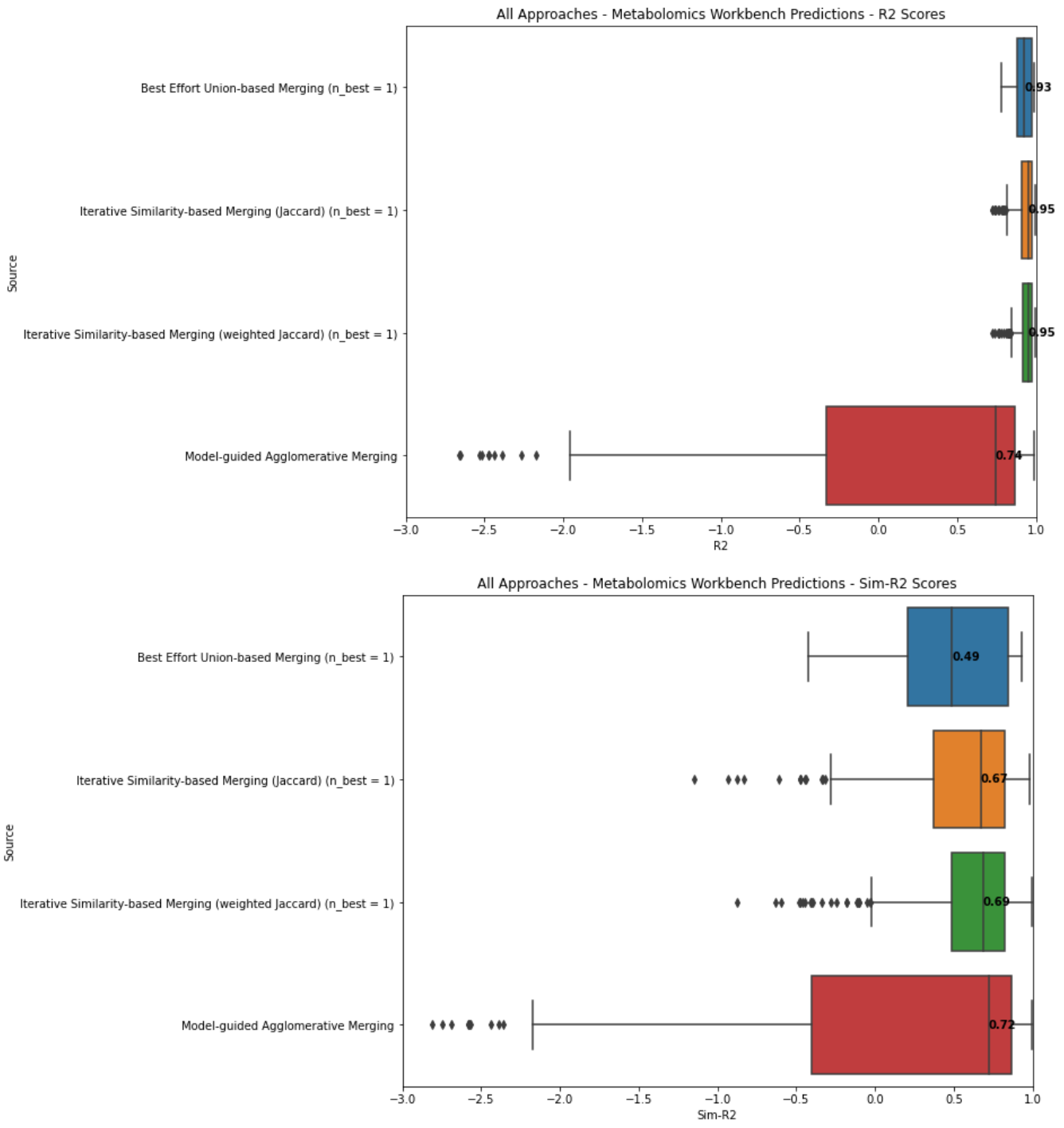

Fig. 8: Performance scores of the baseline union-based merging and the proposed two approaches with best configurations on unseen Metabolomics Workbench data. R^2 at the top and $Sim\text{-}R^2$ at the bottom.

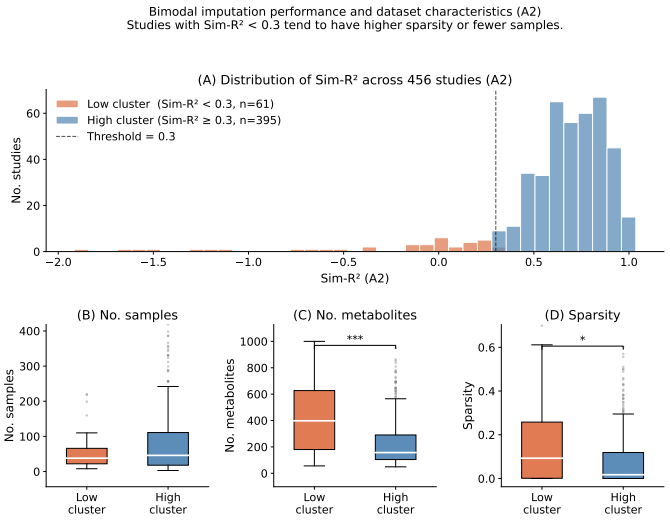

Fig. 9: Distribution of Sim-R^2 scores across Metabolomics Workbench studies for the Model-guided Agglomerative Merging approach. The bimodal distribution reveals two distinct performance clusters: a high-performing group (Sim-$R^2 \geq 0.3$) and a lower-performing group ($n = 61$ studies, Sim-$R^2 < 0.3$), the latter associated with higher metabolite counts (median 398 vs. 157, $p < 0.001$).

non-parametric methods are mathematically and computationally unequipped for cross-dataset integration. Specifically, when subjected to the unified feature space of the Metabolomics Workbench—which exhibits 97.47% sparsity, distance-based metrics (KNN) lose their discriminatory power, and decision-tree algorithms (RF) fail to compute meaningful splits on largely empty features. Therefore, to rigorously evaluate our proposed merging strategies on the task of repository-scale integration, we selected the current state-of-the-art in generative metabolomics modeling (GomariVAE [13]) as our primary baseline, as deep generative architectures are currently the only models capable of bridging such highly sparse, heterogeneous latent spaces.

In particular, we re-implemented the Gomari et al. [13] VAE-based model as a strong generative baseline and extended it with our own union-based merging strategy. The basic merging approach follows the union-based approach with a different VAE model architecture. In their paper, they trained the model on a single large homogeneous TwinsUK dataset [26]. However, since the data was not publicly available, we trained multiple models with the Metabolomics Workbench database based on the proposed *Best Effort Union-based Merging* approach to mimic the idea of creating a large enough dataset to cover more metabolites. Table 4 presents the comparison of state-of-the-art methods on Metabolomics Workbench studies. All the approaches except using the best effort union-based approach have 100% missing metabolite coverage, which means they can predict all missing values in the corresponding studies. Unlike the original Gomari study, which was trained on a single large and homogeneous dataset (TwinsUK), our evaluation was conducted on a highly heterogeneous and sparse dataset collection. As a result, the GomariVAE model exhibited limited generalization capability, yielding an average Sim-R² score of -0.17. This negative score reflects the model's difficulty in reconstructing missing values across diverse datasets with non-overlapping metabolite sets. This outcome underscores the importance of dataset-aware merging strategies, such as our proposed Iterative Similarity-based and Model-guided Agglomerative Merging approaches, which are specifically designed to handle heterogeneity and sparsity in public metabolomics repositories.

We also evaluated our methods using both original and weighted Jaccard similarity metrics to ensure robust comparisons. Our comprehensive evaluation across 437 real-world datasets from the Metabolomics Workbench (filtered from 490 retrieved studies) demonstrates that our proposed approaches consistently outperform these baselines in terms of Sim-R^2, coverage, and scalability.

TABLE 4: Average Sim-R^2 scores of imputation methods on Metabolomics Workbench studies.

Approach	Sim-R^2	Sim-Spearman (median)
GomariVAE (Best Effort Union-based (Jacc.))	-0.17	0.29
Best Effort Union-based (own VAE with Jacc.) ($n_{best} = 1$)	0.49	0.18
Iterative Sim-based (Jacc.) ($n_{best} = 1$)	0.67	0.40
Iterative Sim-based (weighted Jacc.) ($n_{best} = 1$)	0.69	0.40
Model-guided Agglomerative (with 128 joint)	0.72	0.50

4.7 Comparison of Different Similarity Metrics

In this section, we compare the original Jaccard similarity score to our proposed weighted Jaccard similarity scoring scheme. The results are presented in Figure S7 (Suppl. Material). To sum up, the weighted Jaccard score enables us to build more stable and concise models for our model pool, increasing the multi-dataset prediction power.

4.8 Comparison of Training and Prediction Times

In this section, we compare different merging approaches in terms of running times during training and prediction. The training durations for one fold of the proposed merging approaches are presented in Table 5. *Union-based Merging* could not learn and capture any patterns, as it resulted in NaN loss values during its initial training phase. The time requirement for training *Model-guided Agglomerative Merging* was excessive owing to its hierarchical dataset merging structure.

TABLE 5: One-fold training times for different approaches on the Metabolomics Workbench studies.

Approach	Training Time
Union-based	No training
Best Effort Union-based (Jaccard)	28.5 min
Iterative Similarity-based (Jaccard)	43.7 min
Iterative Similarity-based (weighted Jacc.)	48.2 min
Model-guided Agglomerative (with 128 joint)	517.5 min

The prediction times of different approaches were measured based on the average single metabolite prediction time. The results are presented in Table 6. Although configurations with the first similar model, $n_{best} = 1$, provide a fast prediction, its imputation accuracy suffers when compared to *Model-guided Agglomerative Merging*. For larger datasets, *Iterative Similarity-based Merging* with $n_{best} = 1$ configuration provides both a fast and reasonably robust way to

impute missing values, by compromising a small amount of prediction power. However, for small and medium-sized datasets, *Model-guided Agglomerative Merging* is a promising approach.

TABLE 6: Prediction times of one missing metabolite for different approaches trained on Metabolomics Workbench.

Approach	Prediction Time
Union-based	No prediction
Best Effort Union-based (Jacc.) ($n_{best} = 1$)	3.18×10^{-7} s
Best Effort Union-based (Jacc.) ($n_{best} = 5$)	1.28×10^{-6} s
Best Effort Union-based (Jacc.) ($n_{best} = 10$)	3.18×10^{-6} s
Iterative Sim-based (Jacc.) ($n_{best} = 1$)	2.23×10^{-7} s
Iterative Sim-based (Jacc.) ($n_{best} = 5$)	1.21×10^{-6} s
Iterative Sim-based (Jacc.) ($n_{best} = 10$)	2.53×10^{-6} s
Iterative Sim-based (weighted Jacc.) ($n_{best} = 1$)	2.00×10^{-7} s
Iterative Sim-based (weighted Jacc.) ($n_{best} = 5$)	1.16×10^{-6} s
Iterative Sim-based (weighted Jacc.) ($n_{best} = 10$)	2.53×10^{-6} s
Model-guided Agglomerative (with 128 joint)	7.85×10^{-7} s

5 Discussion and Conclusion

In this study, we address the challenge of imputing missing values in large-scale, heterogeneous metabolomics datasets by introducing a novel deep learning-based framework that integrates diverse datasets with non-overlapping metabolite profiles. Our work makes several key contributions that distinguish it from prior approaches:

- We propose two novel dataset merging strategies—**Iterative Similarity-based Merging** and **Model-guided Agglomerative Merging**—that are specifically designed to reduce sparsity while preserving biological diversity across datasets. These strategies enable the construction of imputation-ready datasets that are both dense and representative of real-world variability.
- We introduce a **weighted Jaccard similarity metric** that incorporates both metabolite overlap and value range similarity, improving the selection of merge sets and enhancing model generalizability.
- We develop a **hierarchical ensemble of VAE-based models** that progressively integrates datasets and uses model predictions to fill in missing values during the merging process. This approach allows for robust imputation even in the presence of highly non-overlapping metabolite sets.
- We conduct a comprehensive evaluation on **437 human metabolomics datasets** from the Metabolomics Workbench (filtered from 490 retrieved studies), demonstrating that our methods outperform state-of-the-art baselines in terms of imputation accuracy (Sim-R^2 up to 0.72), coverage, and scalability.

To the best of our knowledge, this is the first study to propose a data source- and dataset-independent imputation framework that is capable of learning from and generalizing across highly heterogeneous metabolomics datasets. To validate the effectiveness of our proposed methods, we conducted a comprehensive comparison against a diverse set of state-of-the-art imputation techniques. These included statistical methods (e.g., KNN, MICE), machine learning-based approaches (e.g., Random Forest), and generative models (e.g., GomariVAE). We selected these baselines based on their prevalence in the metabolomics literature and their demonstrated performance in prior benchmarking studies. Our evaluation, conducted on 437 real-world datasets from the Metabolomics Workbench (filtered from 490 retrieved studies), shows that our methods consistently outperform these baselines in terms of Sim-R^2, coverage, and scalability. This breadth of comparison ensures that our conclusions are robust and grounded in a fair assessment of the current state of the field. Our results show that traditional union-based merging approaches fail to scale due to excessive sparsity, while our proposed methods maintain model trainability and predictive performance.

Recent literature (e.g., [21]) cautions that high proportions of missing data severely compromise imputation accuracy and can distort the compositional nature of metabolomics datasets. Our proposed merging strategies explicitly mitigate this risk. By introducing a strict sparsity threshold (θ) in the Iterative Similarity-based Merging approach, we prevent the creation of overly sparse joint datasets that would otherwise force the model to hallucinate or inflate values. Furthermore, by employing a log_{10} transformation prior to VAE training, our framework helps stabilize the variance inherent in compositional omics data, allowing the model to learn generalizable representations without being biased by extreme outliers.

Furthermore, when interpreting the performance of the baseline integration strategies, it is critical to separate coverage (the proportion of the repository successfully integrated) from imputation accuracy. These methods are not universally applicable to all studies and navigate this trade-off differently:

Standard Union-based Merging: This approach attempts to maximize coverage by forcing all datasets into a single global feature space. However, it resolves non-overlapping metabolites by zero-filling, which inherently generates extreme matrix sparsity (97.47%). This structural penalty severely compromises algorithmic stability and downstream imputation accuracy, rendering it practically ineffective for large-scale integration.

Best-Effort Union Merging: Conversely, this variant sacrifices coverage to preserve accuracy. By strictly limiting the merge to datasets with sufficient natural overlap, it produces viable imputation scores but is only applicable to a small subset of the repository (e.g., 63 out of 437 datasets).

Therefore, direct quantitative comparisons must be contextualized: the Best-Effort variant represents a high-accuracy, low-coverage ceiling, while the Standard Union represents a high-coverage, low-accuracy floor. Our proposed Agglomerative and Iterative approaches are specifically designed to break this dichotomy, achieving both repository-wide coverage and high imputation accuracy without the destructive sparsity penalties of zero-filling.

5.1 Limitations and Future Work

Our proposed approach has several limitations. First, it requires mapping metabolites to a common naming space, which may result in the loss of some dataset-specific information. Additionally, the current framework does not explicitly account for outliers or batch effects.

While our proposed framework demonstrates strong performance across highly heterogeneous merged datasets, we must explicitly qualify the scope of our empirical validation regarding missing mechanisms. The $Sim - R^2$ evaluations conducted in this study rely on simulated random masking. Although deep generative models like VAEs are theoretically well-equipped to model complex, nonlinear Missing Not At Random (MNAR) patterns—such as those commonly arising from instrument Limit of Detection (LOD) thresholds in metabolomics—this specific capability is discussed here as a conceptual advantage. It was not directly validated in our current experimental design. Fully confirming the framework's robustness across all specific missingness types, particularly MNAR, remains an important avenue for future empirical benchmarking.

A further limitation concerns our evaluation protocol: since true ground-truth values for missing metabolites are unavailable by definition, performance is assessed against a pseudo-ground truth generated by KNN imputation on held-out test splits. As a result, reported metrics reflect recovery of KNN-imputed values rather than original instrument measurements, and our Spearman results should be interpreted in this context. This concern is consistent with the findings of Krutkin et al. [21], who show that KNN-based imputation accuracy deteriorates rapidly with increasing missingness rates, further highlighting the inherent limitations of pseudo-ground-truth evaluation in metabolomics.

ACKNOWLEDGMENTS

An extended abstract of this work was presented orally at the 17th International Symposium on Health Informatics and Bioinformatics (18-20 December 2024).

6 DECLARATIONS

6.1 Ethics approval and consent to participate

Not applicable.

6.2 Consent for publication

Not applicable.

6.3 Availability of data and materials

Metabolomics datasets are publicly available at Metabolomics Workbench [32] and MetaboLights [18]. The source codes are available at the following GitHub repository: https://github.com/itu-bioinformatics-database-lab/vertical_imputer.

6.4 Competing interests

None.

6.5 Funding

This work was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) [grant number 124N069] and the National Center for High Performance Computing (UHEM) [Grant Number: 1009742021].

6.6 Authors' contributions

SC implemented the ML models and performed evaluations, BC curated the datasets, AC conceived the study and acquired the funding. All authors contributed to the manuscript's writing.

REFERENCES

[1] E. G. Armitage, J. Godzien, V. Alonso-Herranz, C. Barbas, and A. Lopez-Gonzalvez. Missing value imputation strategies for metabolomics data. *ELECTROPHORESIS*, 36(24):3050–3060, 2015.

[2] A. Cakmak and M. H. Celik. Personalized metabolic analysis of diseases. *IEEE/ACM Trans Comput Biol Bioinform*, 18(3):1014–1025, June 2021.

[3] P. Cui, H. Wang, and Z. Bai. Integrated single-cell and bulk rna-seq analysis identifies a prognostic t-cell signature in colorectal cancer. *Scientific Reports*, 14(1):20177, 2024.

[4] H. De los Santos, K. P. Bennett, and J. M. Hurley. Mosaic: a joint modeling methodology for combined circadian and non-circadian analysis of multi-omics data. *Bioinformatics*, 37(6):767–774, 2021.

[5] J. P. Dekermanjian, E. Shaddox, D. Nandy, D. Ghosh, and K. Kechris. Mechanism-aware imputation: a two-step approach in handling missing values in metabolomics. *BMC Bioinformatics*, 23(1):179, May 2022.

[6] K. T. Do, S. Wahl, J. Raffler, S. Molnos, M. Laimighofer, J. Adamski, K. Suhre, K. Strauch, A. Peters, C. Gieger, C. Langenberg, I. D. Stewart, F. J. Theis, H. Grallert, G. Kastenmüller, and J. Krumsiek. Characterization of missing values in untargeted ms-based metabolomics data and evaluation of missing data handling strategies. *Metabolomics*, 14(10):128, Sep 2018.

[7] J.-H. Du, Z. Cai, and K. Roeder. Robust probabilistic modeling for single-cell multimodal mosaic integration and imputation via scvaeit. *Proceedings of the National Academy of Sciences*, 119(49):e2214414119, 2022.

[8] E. Fahy and S. Subramaniam. Refmet: a reference nomenclature for metabolomics. *Nature Methods*, 17(12):1173–1174, Dec 2020.

[9] T. Faquih, M. van Smeden, J. Luo, S. le Cessie, G. Kastenmüller, J. Krumsiek, R. Noordam, D. van Heemst, F. R. Rosendaal, A. van Hylckama Vlieg, K. Willems van Dijk, and D. O. Mook-Kanamori. A workflow for missing values imputation of untargeted metabolomics data. *Metabolites*, 10(12), 2020.

[10] A. Fouché, L. Chadoutaud, O. Delattre, and A. Zinovyev. Transmorph: a unifying computational framework for modular single-cell rna-seq data integration. *NAR Genomics and Bioinformatics*, 5(3):lqad069, 2023.

[11] C. Fu, J. Lu, N. Zhao, and W. Ling. Mosaic: A pipeline for microbiome studies analytical integration and correction. *bioRxiv*, 2024.

[12] S. Ghazanfar, C. Guibentif, and J. C. Marioni. Stabilized mosaic single-cell data integration using unshared features. *Nature biotechnology*, 42(2):284–292, 2024.

[13] D. P. Gomari, A. Schweickart, L. Cerchietti, E. Paietta, H. Fernandez, H. Al-Amin, K. Suhre, and J. Krumsiek. Variational autoencoders learn transferrable representations of metabolomics data. *Communications Biology*, 5(1):645, Jun 2022.

[14] L. Gondara and K. Wang. Mida: Multiple imputation using denoising autoencoders, 2018.

[15] Z. He, S. Hu, Y. Chen, S. An, J. Zhou, R. Liu, J. Shi, J. Wang, G. Dong, J. Shi, et al. Mosaic integration and knowledge transfer of single-cell multimodal data with midas. *Nature Biotechnology*, pages 1–12, 2024.

[16] O. Hrydziuszko and M. Viant. Missing values in mass spectrometry based metabolomics: An undervalued step in the data processing pipeline. *Metabolomics*, 8:161–174, 06 2011.

[17] Z. Jin, J. Kang, and T. Yu. Missing value imputation for LC-MS metabolomics data by incorporating metabolic network and adduct ion relations. *Bioinformatics*, 34(9):1555–1561, 12 2017.

[18] N. S. Kale, K. Haug, P. Conesa, K. Jayseelan, P. Moreno, P. Rocca-Serra, V. C. Nainala, R. A. Spicer, M. Williams, X. Li, R. M. Salek, J. L. Griffin, and C. Steinbeck. MetaboLights: An Open-Access database repository for metabolomics data. *Curr Protoc Bioinformatics*, 53:14.13.1–14.13.18, Mar. 2016.

[19] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *CoRR*, abs/1906.02691, 2019.

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes, 2022.

[21] D. D. Krutkin, S. P. Thomas, S. Zuffa, P. Rajkumar, R. Knight, P. C. Dorrestein, and S. T. Kelley. To impute or not to impute in untargeted metabolomics—that is the compositional question. *Journal of the American Society for Mass Spectrometry*, 36(4):742–759, 2025.

[22] N. Kumar, M. A. Hoque, M. Shahjaman, S. S. Islam, and M. N. Haque Mollah. A new approach of outlier-robust missing value imputation for metabolomics data analysis. *Current Bioinformatics*, 14(1):43–52, 2019.

[23] M. Loza, S. Teraguchi, D. M. Standley, and D. Diez. Unbiased integration of single cell transcriptome replicates. *NAR genomics and bioinformatics*, 4(1):lqac022, 2022.

[24] A. Maćkiewicz and W. Ratajczak. Principal components analysis (pca). *Computers and Geosciences*, 19(3):303–342, 1993.

[25] E. M. Mirkes, J. Bac, A. Fouché, S. V. Stasenko, A. Zinovyev, and A. N. Gorban. Domain adaptation principal component analysis: base linear method for learning with out-of-distribution data. *Entropy*, 25(1):33, 2022.

[26] A. Moayyeri, C. J. Hammond, D. J. Hart, and T. D. Spector. The UK adult twin registry (TwinsUK resource). *Twin Res Hum Genet*, 16(1):144–149, Oct. 2012.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, volume 32, pages 8024–8035. Curran Associates, Inc., 2019.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[29] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks — ICANN'97*, pages 583–588, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

[30] D. J. Stekhoven and P. Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011.

[31] D. J. Stekhoven and P. Bühlmann. MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 10 2011.

[32] M. Sud, E. Fahy, D. Cotter, K. Azam, I. Vadivelu, C. Burant, A. Edison, O. Fiehn, R. Higashi, K. S. Nair, S. Sumner, and S. Subramaniam. Metabolomics workbench: An international repository for metabolomics data and metadata, metabolite standards, protocols, tutorials and training, and analysis tools. *Nucleic Acids Res*, 44(D1):D463–70, Oct. 2015.

[33] D. Sun, X. Guan, A. E. Moran, L.-Y. Wu, D. Z. Qian, P. Schedin, M.-S. Dai, A. V. Danilov, J. J. Alumkal, A. C. Adey, et al. Identifying phenotype-associated subpopulations by integrating bulk and single-cell sequencing data. *Nature biotechnology*, 40(4):527–538, 2022.

[34] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays . *Bioinformatics*, 17(6):520–525, 06 2001.

[35] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays . *Bioinformatics*, 17(6):520–525, 06 2001.

[36] S. van Buuren and K. C. Oudshoorn. *Flexible Multivariate Imputation by MICE*. TNO Prevention and Health, 1999.

[37] T. B. Verhey, H. Seo, A. Gillmor, V. Thoppey-Manoharan, D. Schriemer, and S. Morrissy. mosaicmpi: a framework for modular data integration across cohorts and -omics modalities. *Nucleic Acids Research*, 52(12):e53–e53, 05 2024.

[38] R. Wei, J. Wang, M. Su, E. Jia, S. Chen, T. Chen, and Y. Ni. Missing value imputation approach for mass spectrometry-based metabolomics data. *Scientific Reports*, 8(1):663, Jan 2018.

[39] J. Xu, Y. Wang, X. Xu, K.-K. Cheng, D. Raftery, and J. Dong. Nmf-based approach for missing values imputation of mass spectrometry metabolomics data. *Molecules*, 26(19), 2021.

[40] T.-L. Yang, H. Shen, A. Liu, S.-S. Dong, L. Zhang, F.-Y. Deng, Q. Zhao, and H.-W. Deng. A road map for understanding molecular and genetic determinants of osteoporosis. *Nature Reviews Endocrinology*, 16(2):91–103, Feb 2020.

[41] J. Yoon, J. Jordon, and M. van der Schaar. Gain: Missing data imputation using generative adversarial nets, 2018.

[42] Z. Zhang, H. Sun, R. Mariappan, X. Chen, X. Chen, M. S. Jain, M. Efremova, S. A. Teichmann, V. Rajan, and X. Zhang. scmomat jointly performs single cell mosaic integration and multi-modal bio-marker detection. *Nature Communications*, 14(1):384, 2023.

[43] C. Zhao, K.-J. Su, C. Wu, X. Cao, Q. Sha, W. Li, Z. Luo, T. Qin, C. Qiu, L. J. Zhao, A. Liu, L. Jiang, X. Zhang, H. Shen, W. Zhou, and H.-W. Deng. Multi-view variational autoencoder for missing value imputation in untargeted metabolomics. *ArXiv*, Oct. 2023.

[44] S. Zheng and N. Charoenphakdee. Diffusion models for missing value imputation in tabular data, 2023.