Missing view completion for multi-timepoints omics data via representation disentanglement and temporal knowledge transfer

Siyu Han[1,2,3], Shixiang Yu[1,2,3], Mengya Shi[1,2,3], Makoto Harada[2,3], Jianhong Ge[2,3], Jiesheng Lin[4,5], Cornelia Prehn[6], Agnese Petrera[6], Ying Li[7], Flora Sam[8,9], Giuseppe Matullo[10], Jerzy Adamski[11,12,13], Karsten Suhre[14], Christian Gieger[4,15], Stefanie M. Hauck[6], Christian Herder[16,17,18], Michael Roden[16,17,18], Francesco Paolo Casale[19,20], Na Cai[1,19], Annette Peters[3,4,5,21], Rui Wang-Sattler[2,3,5,*]

[1]TUM School of Medicine, Technical University of Munich, Munich, Germany

[2]Institute of Translational Genomics, Helmholtz Zentrum München, German Research Center for Environmental Health, Neuherberg, Germany

[3]German Center for Diabetes Research (DZD), Partner Neuherberg, Neuherberg, Germany

[4]Institute of Epidemiology, Helmholtz Zentrum München, German Research Center for Environmental Health, Neuherberg, Germany

[5]Institute for Medical Information Processing, Biometry, and Epidemiology (IBE), Faculty of Medicine, Ludwig-Maximilians-Universität München, Pettenkofer School of Public Health, Munich, Germany

[6]Metabolomics and Proteomics Core, Helmholtz Zentrum München, German Research Center for Environmental Health, Neuherberg, Germany

[7]College of Computer Science and Technology, Key Laboratory of Symbol Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, China

[8]Eli Lilly and Company, Lilly Corporate Center, Indianapolis, IN, USA

[9]Whitaker Cardiovascular Institute, Boston University Chobanian & Avedisian School of Medicine, Boston, MA, USA

[10]Genomics Variation, Complex Diseases and Population Medicine Unit, Turin University, Italy

[11]Institute of Experimental Genetics, Helmholtz Zentrum München, German Research Center for Environmental Health, Neuherberg, Germany

[12]Department of Biochemistry, Yong Loo Lin School of Medicine, National University of Singapore, Singapore.

[13]Institute of Biochemistry, Faculty of Medicine, University of Ljubljana, Ljubljana, Slovenia

[14]Department of Physiology and Biophysics, Weill Cornell Medicine - Qatar, Education City - Qatar Foundation, Doha, Qatar

[15]Research unit of Molecular Epidemiology, Helmholtz Zentrum München, German Research Center for Environmental Health, Neuherberg, Germany

[16]Institute for Clinical Diabetology, German Diabetes Center, Leibniz Center for Diabetes Research at Heinrich-Heine-University Düsseldorf, Düsseldorf, Germany

[17]German Center for Diabetes Research (DZD), Partner Düsseldorf, Neuherberg, Germany

[18]Department of Endocrinology and Diabetology, Medical Faculty and University Hospital Düsseldorf, Heinrich-Heine-University Düsseldorf, Düsseldorf, Germany

[19]Helmholtz Pioneer Campus, Helmholtz Zentrum München, German Research Center for Environmental Health, Neuherberg, Germany

[20]TUM School of Computation, Information and Technology (CIT), Technical University of Munich, Munich, Germany

[21]Munich Heart Alliance, German Center for Cardiovascular Health (DZHK E.V., Partner-Site Munich), Munich, Germany

Corresponding author: Rui Wang-Sattler, E-mail: rui.wang-sattler@helmholtz-munich.de

Abstract

Longitudinal multi-view omics data offer unique insights into the temporal dynamics of individual-level physiology that is instrumental to advancements in personalized healthcare. However, the common occurrence of incomplete view makes inference tasks difficult, and there is a lack of tailored methods for solving this critical issue. Here, we introduce LEOPARD, an approach specifically designed to complete missing views for multi-timepoint omics data. By disentangling longitudinal omics data into content and temporal representations, LEOPARD transfers temporal knowledge to omics-specific content to complete missing views. Compared to generic imputation methods, LEOPARD gives the most robust and reliable results across three longitudinal proteomics and metabolomics datasets. LEOPARD-imputed data also achieve the closest performance to observed data in our regression and classification analyses. Our work takes the first step towards a principled and generalized treatment of missing views in longitudinal omics data, enabling comprehensive exploration of temporal changes within omics data.

Introduction

The rapid advancement of omics technologies has enabled researchers to obtain high-dimensional datasets across multiple modalities, providing unprecedented insights into various diseases[1]. Each view contributes a partial or entirely independent perspective on complex biological systems[2]. While advancements in multi-omics measurements have increased throughput and enabled the acquisition of multiple modalities in a single assay[3], data preprocessing, analysis, and interpretation remain significant and important challenges.

One of the most pressing challenges is the presence of missing data[4], which at its best (when missingness occurs at random) may reduce statistical power, and at its worst (when it is not random) can lead to biased discoveries. In longitudinal studies that can span decades, the problem of missing views in multi-timepoint omics data becomes increasingly common due to factors such as dropout in omics measurements, experimental errors, or unavailability of specific omics profiling platforms at certain timepoints. The incompleteness of these datasets hinders multi-omics integration[5] and investigations into predisposing factors (such as age and genetics), enabling factors (such as healthcare service and physical activity), and biomarkers for diseases.

Missing view completion refers to the estimation of missing omics data in a multi-view context, where each view could correspond to different types of omics data or data acquired through various platforms. As shown in Fig. 1 (a), the task of missing view completion is more complex than cross-sectional missing value imputation, as it needs to harmonize data distributions across views[6] and capture

temporal patterns[7]. Generic methods, such as mice (Multivariate Imputation by Chained Equations)[8], missForest[9], and KNNimpute[10], learn direct mappings between views from observed data without missingness. However, this strategy is inadequate or suboptimal[11] for longitudinal data. The learned mappings may overfit the training timepoints, making them unsuitable for inferring data at other timepoints, especially when biological variations cause distribution shifts over time (Fig. 1 (b)). Further, this precludes investigations into temporal variation, which can be of great interest. Given these limitations, there is an increasing demand for view completion methods suitable for longitudinal data. Regarding evaluation of imputation results, while metrics such as mean squared error (MSE) and percent bias (PB) are commonly used to evaluate imputation results[22], these quantitative metrics alone may not fully capture data quality in the context of omics data. As depicted in Fig. 1 (c), data imputed by method 1 may have a lower MSE than that imputed by method 2, but at a loss of biologically meaningful variations. Further case studies would be helpful to evaluate the quality of imputed data.

In this paper, we introduce LEOPARD (missing view comp**l**etion for multi-tim**e**points **o**mics data via re**p**resentation disent**a**nglement and tempo**r**al knowle**d**ge transfer), a neural network-based approach that offers a novel and effective solution to this challenge (Fig.1 (d)). LEOPARD extends representation disentanglement[12] and style transfer[13] techniques, which have been widely applied in various contexts such as image classification[14], image synthesis[15], and voice conversion[16], to the application of missing view completion in longitudinal omics data. LEOPARD factorizes omics data from different timepoints into omics-specific content and timepoint-specific knowledge via contrastive learning. Missing views are completed by transferring temporal knowledge to the corresponding omics-specific content.

We demonstrate the effectiveness of LEOPARD through extensive simulations using human proteomics and metabolomics data from the MGH (Massachusetts General Hospital) COVID study[17] and the KORA (Cooperative Health Research in the Region of Augsburg) cohort[18–21] (Fig. 1 (e)). Additionally, we perform two case studies to assess whether biological information is preserved in the imputed data, providing a comprehensive assessment of LEOPARD's performance in both regression and classification tasks using real omics data (Fig.1 (f)).

In summary, the key contributions of this study are:

- We propose LEOPARD, a novel method tailored for missing view completion in multi-timepoint omics data that innovatively applies representation disentanglement and style transfer.

- Our study shows that generic imputation methods designed for cross-sectional data are not suitable for longitudinal data, emphasizing the need for tailored approaches. Additionally, we highlight that canonical evaluation metrics do not adequately reflect the quality of imputed biomedical data. Further investigations including regression and classification analyses can augment these metrics in assessment of imputation quality and preservation of biological variations.
- Our research reveals that omics data across timepoints can be factorized into content and temporal knowledge, providing a foundation for further explorations into the temporal dynamics of biological. This insight offers a novel perspective for predictive healthcare that extends beyond the problem of data imputation.

Problem Formulation

In our missing view completion problem, we are given a generalized dataset $\mathcal{X}_{V,T}^{N}$ that includes omics data from N individuals across multiple views (V) and timepoints (T). For simplification, we consider data of two views $\{v1, v2\}$ from two timepoints $\{t1, t2\}$. Data in $\mathcal{X}_{V=v2,T=t2}^{N}$ are assumed as incomplete. The dataset is split into training, validation, and test sets, denoted as $\mathcal{D}_{v,t}^{split}$. The task is to complete the missing view of $\mathcal{D}_{v=v2,t=t2}^{\text{test}}$. To achieve this, we attempt to develop a model that can not only capture the mappings from $v1$ to $v2$, but also extract temporal knowledge of $t1$ and $t2$ using the available observed data in the training set.

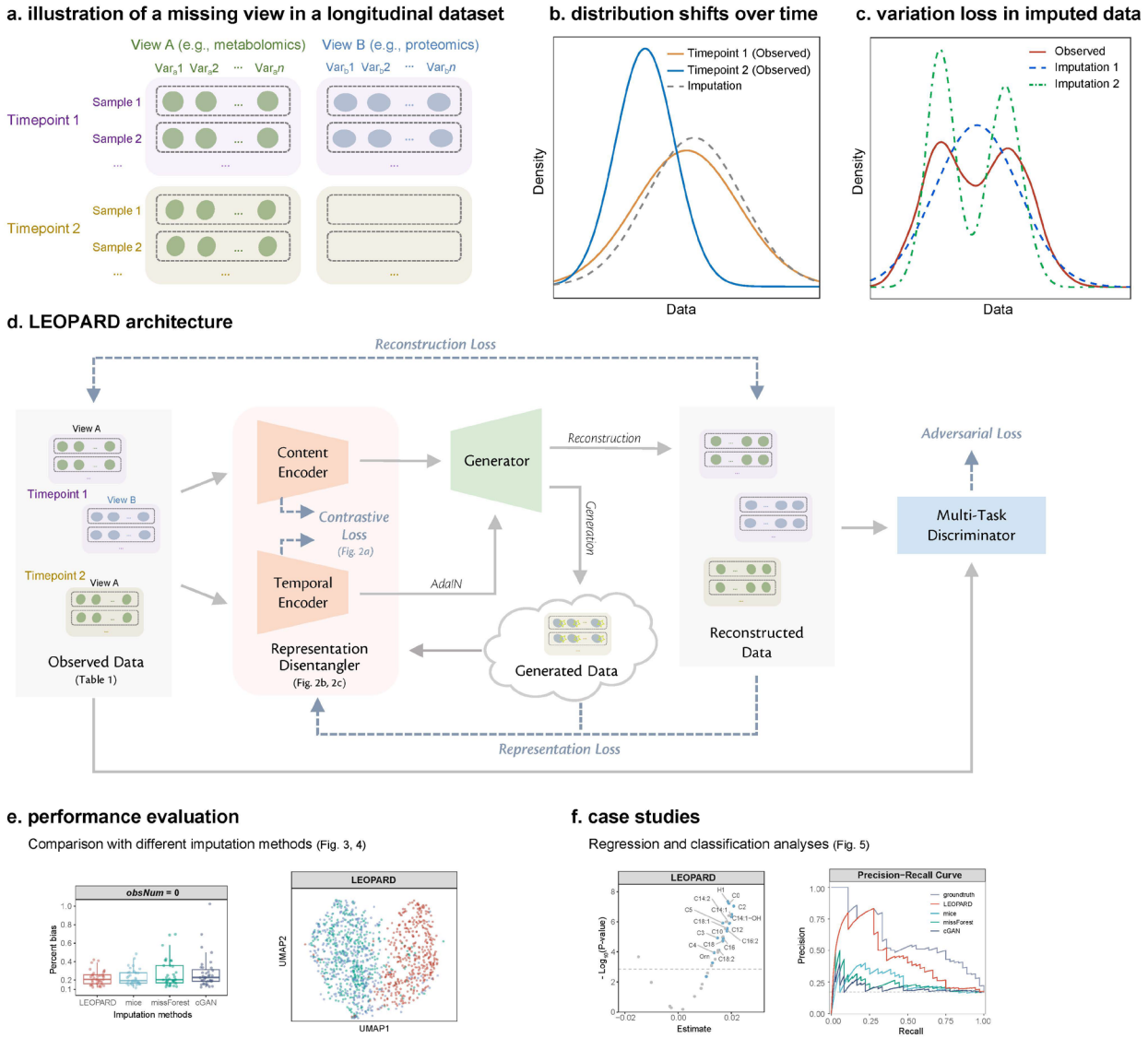


Fig. 1. Problem description and overview of LEOPARD architecture.

a, An example of a missing view in a longitudinal multi-omics dataset. In this instance, View A and View B correspond to metabolomics and proteomics data, respectively. Data of View B at Timepoint 2 are absent. **b**, An example of data density calculated from raw data (Timepoint 1 and 2) and imputed data. The imputation model built from one timepoint (Timepoint 1) might not be appropriate for inferring data from another timepoint (Timepoint 2). **c**, Compared to Raw data, data of Imputation 1 may exhibit lower MSE than data of Imputation 2, but Imputation 1 potentially lose biological variations present in the data. **d**, The architecture of LEOPARD. Omics data from multiple timepoints are disentangled into omics-specific content representation and temporal knowledge by the content and temporal encoders. The generator learns mappings between two views, while temporal knowledge is injected into content representation via the AdaIN operation. The multi-task discriminator encourages the distributions of reconstructed data to align more closely with the actual distribution. Contrastive loss enhances the representation learning process. Reconstruction loss measures the MSE between the input and reconstructed data. Representation loss stabilizes the training process by minimizing the MSE between the same kinds of content and temporal representations across timepoints and views. Adversarial loss is incorporated to alleviate the element-wise averaging issue of the MSE loss. **e**, the performance of LEOPARD is evaluated with PB and UMAP embeddings. **f**, two case studies includes both regression and classification analyses are performed to evaluate if biological information is preserved in the imputed data.

Results

Table 1. Summary of benchmark datasets used in this study

	MGH COVID proteomics	KORA metabolomics	KORA multi-omics
Omics type	proteomics	metabolomics	metabolomics; proteomics
View			
$v1$	panel: Target 96 cardiometabolic	biochemical class*: GPL	omics: metabolomics
$v2$	panel: Target 96 inflammation	biochemical class*: AC, SL, AA, MS	omics: proteomics
Variable number	Target 96 cardiometabolic: 79, Target 96 inflammation: 66	GPL: 70, AC: 15, SL: 11, AA: 9, MS: 1	metabolites: 104, proteins: 66
Timepoint			
$t1$	D0	F4	S4
$t2$	D3	FF4	F4
Time span	3 days	7 years	7 years
Sample number			
Total (100%)	218	2,085	1,062
Training (64%)	140	1,335	680
Validation (16%)	35	333	170
Test (20%)	43	417	212
Masked data for completion	Target 96 inflammation proteins from D3	non-GPL metabolites from FF4	proteins from F4

*The targeted metabolites from KORA are classified into five analyte classes, namely glycerophospholipid (GPL), acylcarnitine (AC), sphingolipid (SL), amino acid (AA), and monosaccharide (MS). One view group comprises 70 metabolites from the glycerophospholipid class, while the other view group comprises 36 metabolites from the other classes.

Data Preparation

Three distinct datasets are designed based on data variations, time spans, and sample sizes (Table 1). The first two are mono-omics datasets, constructed with the proteomics data from MGH COVID study and the metabolomics from the KORA cohort, respectively. In the mono-omics datasets, views correspond to panels or biochemical classes, simulating the common issues encountered in longitudinal studies where data from certain panels or biochemical classes are incomplete. The third dataset is a multi-omics dataset constructed with both metabolomics and proteomics data from the KORA cohort. In this dataset, views correspond to different omics types. As described in Problem Statement section, each dataset comprises data of two views (or view groups, $v1$ and $v2$) from two timepoints ($t1$ and $t2$), where $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$ is masked for performance evaluation.

Dataset from MGH COVID study The MGH COVID study includes plasma proteins measured by the Olink® Explore 1536 platform for patients at three timepoints: on day 0 (D0) for all patients on the day they were hospitalized for COVID, and on days 3 (D3) and 7 (D7) for patients still hospitalized then. In this study, we utilize proteomics data from D0 ($t1$) and D3 ($t2$), which

have the largest sample sizes (N = 218). The two views in this dataset are constructed with proteins passing quality control (QC, Methods) in the Olink® Target 96 Cardiometabolic (79 proteins, $v1$) and Inflammation assays (66 proteins, $v2$).

Datasets from the KORA cohort Data from the KORA study are extracted from the baseline survey (S4, examined between 1999 and 2001), the first follow-up study (F4, 2006~2008), and the second follow-up study (FF4, 2013~2014). Metabolomics data are available at all three timepoints, though the analytical kits used are different between timepoints: data from S4 and FF4 were measured using AbsoluteIDQ® p180 (p180), while data from F4 were measured with AbsoluteIDQ® p150 (p150). Proteomics data are available at two timepoints, S4 and F4, both measured using the Olink® Target 96 Inflammation assay.

We use metabolomics data (N = 2,085) from F4 ($t1$) and FF4 ($t2$) to construct the KORA metabolomics dataset. After QC[23–25] (Methods), 106 targeted metabolites are selected from five analyte classes, namely acylcarnitine (AC), amino acid (AA), glycerophospholipid (GPL), sphingolipid (SL), and monosaccharide (MS). The metabolites are divided into two view groups: 70 metabolites from GPL ($v1$) and 36 metabolites from the other four classes ($v2$).

We use metabolomics and proteomics data (N = 1,062) from S4 ($t1$) and F4 ($t2$) to construct the KORA multi-omics dataset. TIGER (Technical variation elImination with ensemble learninG architEctuRe)[26] is employed to remove technical effects introduced by different analytical kits (Methods). Following QC, we obtain 104 metabolites ($v1$) and 66 proteins ($v2$).

CGAN Architecture as a baseline method

Existing neural network-based missing view completion methods[27–29] have shown remarkable performance in the field of computer vision. Given their inapplicability to omics data, we constructed a conditional generative adversarial network (cGAN)-based architecture[30] specifically tailored for omics data, as a baseline method. This architecture is inspired by VIGAN (View Imputation via Generative Adversarial Networks)[31] and a method proposed by Cai *et al.*[32], both initially designed for multi-modality image completion.

In the training phase, the generator of the cGAN is trained on $\mathcal{D}^{\text{train}}_{v=v1,t=t1}$ and $\mathcal{D}^{\text{train}}_{v=v2,t=t1}$ to capture the mappings between two views. The discriminator guides the generator to produce data with a distribution similar to that of actual data. The discriminator also has an auxiliary classifier[33] to ensure generated data can be paired with input data (Methods). In the inference phase, the generator utilizes the mappings it has learned from $t1$ to impute the missing view at

$t2$. Compared to popular methods such as mice and missForest, our cGAN model has the potential to learn more complex mappings between views. This cGAN model exemplifies the performance of existing neural network-based cross-sectional methods when applied to longitudinal cases.

Overview of LEOPARD architecture

As illustrated in Fig. 1 (d), the LEOPARD architecture comprises several hierarchical components. First, data of each view are transformed into vectors of equal dimensions using corresponding pre-layers. Subsequently, omics data are decomposed into content and temporal representations. The content encoder captures the intrinsic content of the views, while the temporal encoder extracts knowledge specific to different timepoints. A generator then reconstructs observed views or completes missing views by transferring the temporal knowledge to the view-specific content using Adaptive Instance Normalization (AdaIN)[13]. Lastly, a multi-task discriminator is used to discriminate between real and generated data (Methods).

The LEOPARD model is trained by minimizing four types of losses: contrastive loss, representation loss, reconstruction loss, and adversarial loss. Minimizing normalized temperature-scaled cross-entropy (NT-Xent)-based contrastive loss[34] optimizes the factorization of data into content and temporal representation. For both representations, minimizing the contrastive loss brings together the data pairs from the same view or timepoint and pushes apart the data pairs from different ones, so that the encoders learn similar intrinsic content (or temporal knowledge) across timepoints (or views). The representation loss, also computed on content and temporal representations, measures the MSE of the representations factorized from the actual and reconstructed data. LEOPARD minimizes this loss based on the intuition that the representations of the actual and reconstructed data should be alike. The reconstruction loss measures the MSE between imputed and observed values. Previous studies[35–37] have demonstrated that the optimization of MSE loss often results in averaged outputs, leading to blurring effects when generating images. In our context, this might diminish biological variations present in omics data. To alleviate this issue, we use adversarial loss to encourage the predicted distribution to align more closely with the actual distribution.

LEOPARD has three unique features compared to conventional architectures for multi-view data completion. First, instead of focusing on direct mappings between views, which can only be learned from paired data where both views are present, LEOPARD formulates this imputation task in terms of representation learning and style transfer. This allows LEOPARD to utilize all

available data, including observations present in only one view or timepoint. Second, it incorporates contrastive loss to disentangle the representations unique to views and timepoints, which enables the model to learn more generalized and structured representations. Our experiments show that this is of importance to improve data quality. Third, its multi-task discriminator solves multiple adversarial classification tasks simultaneously by yielding multiple binary prediction results, which is more efficient and effective than a discriminator for a multi-class classification problem[38] or several discriminators for different views and timepoints.

The representation disentanglement of LEOPARD

We use the KORA multi-omics dataset to examine if LEOPARD can effectively disentangle content and temporal representations from omics data. The model is trained for 600 epochs to impute $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$, with the uniform manifold approximation and projection (UMAP)[39] employed for visualizing the content and temporal representations of the validation set across different views and timepoints. As the training progresses, we expect similar representations gradually cluster together, while dissimilar ones form distinct clusters in the UMAP plot.

As the model trains, the contrastive loss decreases (Fig. 2 (a)), indicating that LEOPARD is increasingly able to encode the representations for different views and timepoints. The content representation embeddings (Fig. 2 (b)) of observed $v1$ (in blue) and $v2$ (in red) separate rapidly during training (epoch 5), but those of the imputed $v2$ for $t2$ (in light red) do not mix with those of the observed $v2$ at $t1$ (in dark red). This suggests that while LEOPARD can distinguish between $v1$ and $v2$ after only a few training epochs, it is not yet capable of producing high-quality $v2$ for $t2$ that have similar content information as the observed $v2$ at $t1$. After 30 epochs of training, the content representation of $v2$ at $t2$ is better encoded, with its embeddings mixing with those of $v2$ at $t1$. Similar trends are observed in the temporal representations (Fig. 2 (c)), where embeddings of each timepoint ($t1$ and $t2$ in dark and light colors, respectively) gradually form clusters as training progresses. We notice that the temporal representations take more epochs than the content representations to form distinct clusters. Even after 100 epochs, some temporal representation embeddings of $t1$ are still mixed with those of $t2$. However, after around 450 epochs, LEOPARD is demonstrably able to encode temporal information that is unique to $t1$ and $t2$ (Fig. 2 (c)).

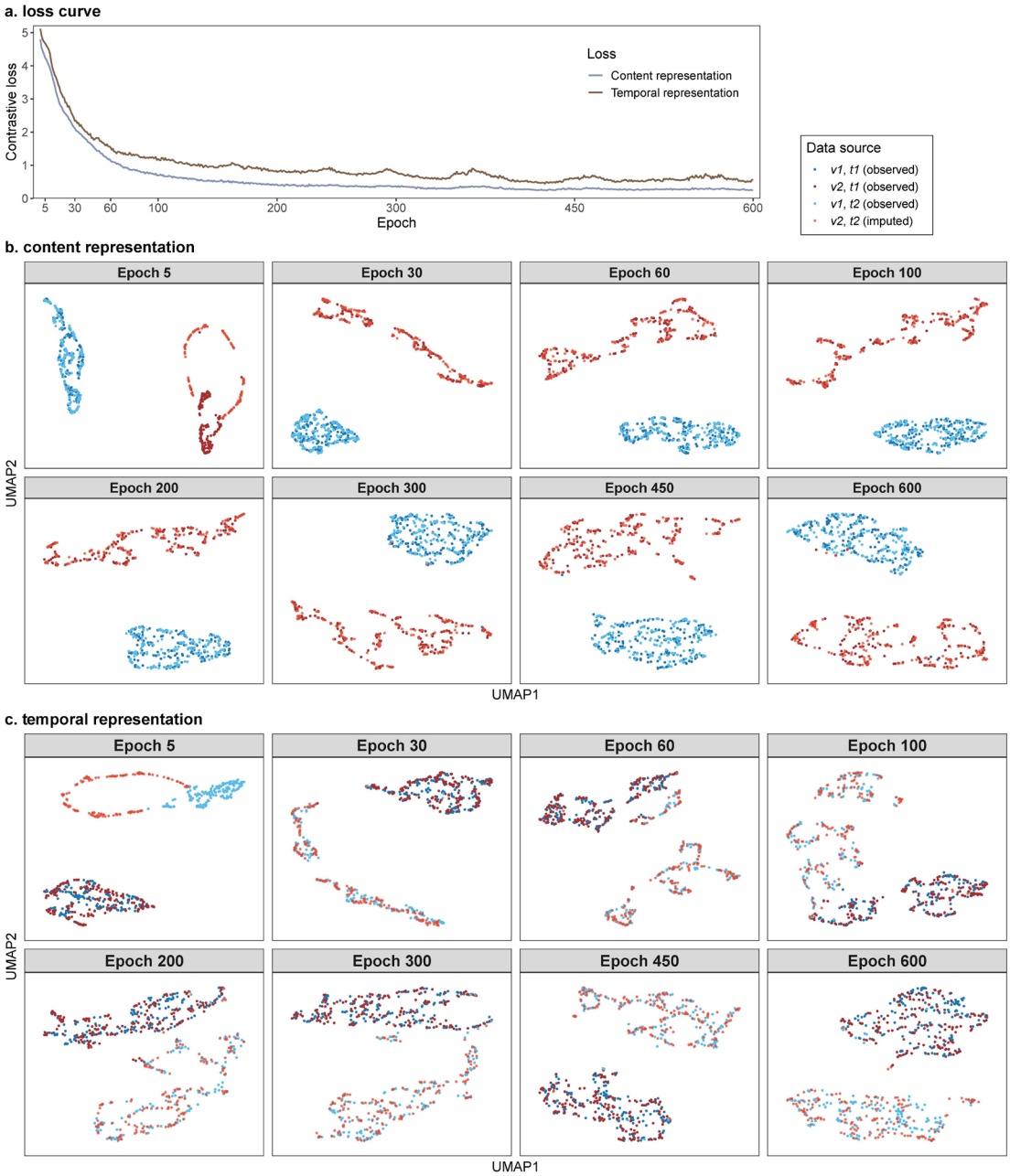


Fig. 2. Contrastive loss and UMAP embeddings illustrate the representation disentanglement process of LEOPARD on the KORA multi-omics dataset.

a, The NT-Xent-based contrastive loss is computed for content and temporal representations. **b-c**, UMAP embeddings of content (**b**) and temporal (**c**) representations at various training epochs are visualized for the KORA multi-omics dataset's validation set. Representations encoded from data of $v1$ and $v2$ (metabolomics and proteomics, depicted by blue and red dots) at timepoint $t1$ and $t2$ (S4 and F4, depicted by dark- and light-colored dots) are plotted. The data of $v2$ at $t2$ are imputed data produced after each training epoch, while the other data are from the observed samples in the validation set. LEOPARD's content and temporal encoders capture signals unique to omics-specific content and temporal variations. In (**b**), as the training progresses, one cluster is formed by the data of $v2$ at $t1$ and $t2$ (dark and light red dots), while the other cluster is formed by the data of $v1$ at $t1$ and $t2$ (dark and light blue dots), indicating that the content encoder is able to encode timepoint-invariant content representations. Similarly, in (**c**), embeddings from the same timepoint cluster together. One cluster is formed by the data of $v1$ and $v2$ at $t1$ (dark blue and red dots), and the other is formed by the data of $v1$ and $v2$ at $t2$ (light blue and red dots). This demonstrates that LEOPARD can effectively factorize omics data into content and temporal representations.

Evaluating LEOPARD and other popular methods on the benchmark datasets

We use two evaluation metrics, PB and UMAP visualization, to assess the performance of mice, missForest, cGAN, and LEOPARD on $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$ of each benchmark dataset. PB quantifies the median absolute error ratio between the observed and imputed values for each variable, offering a variable-level assessment of imputation performance. In contrast, UMAP visualization illustrates overall similarities between observed and imputed datasets, providing a dataset-level evaluation.

During training, the methods mice, missForest, and cGAN only support the data where both views are observed. These data include $\mathcal{D}^{\text{train}}_{v=v1,t=t1}$ and $\mathcal{D}^{\text{train}}_{v=v2,t=t1}$, along with different numbers of training observations (*obsNum*) from $\mathcal{D}^{\text{train}}_{v=v1,t=t2}$ and $\mathcal{D}^{\text{train}}_{v=v2,t=t2}$. By varying *obsNum*, we assess how additional observed data from $t2$ affects imputation performance. In contrast, as LEOPARD is able to learn from training observations where only one view is available, the entire $\mathcal{D}^{\text{train}}_{v=v1,t=t2}$ is included in its training. The variation of *obsNum* only affects the number of observed samples from $\mathcal{D}^{\text{train}}_{v=v2,t=t2}$ used for training LEOPARD (Methods).

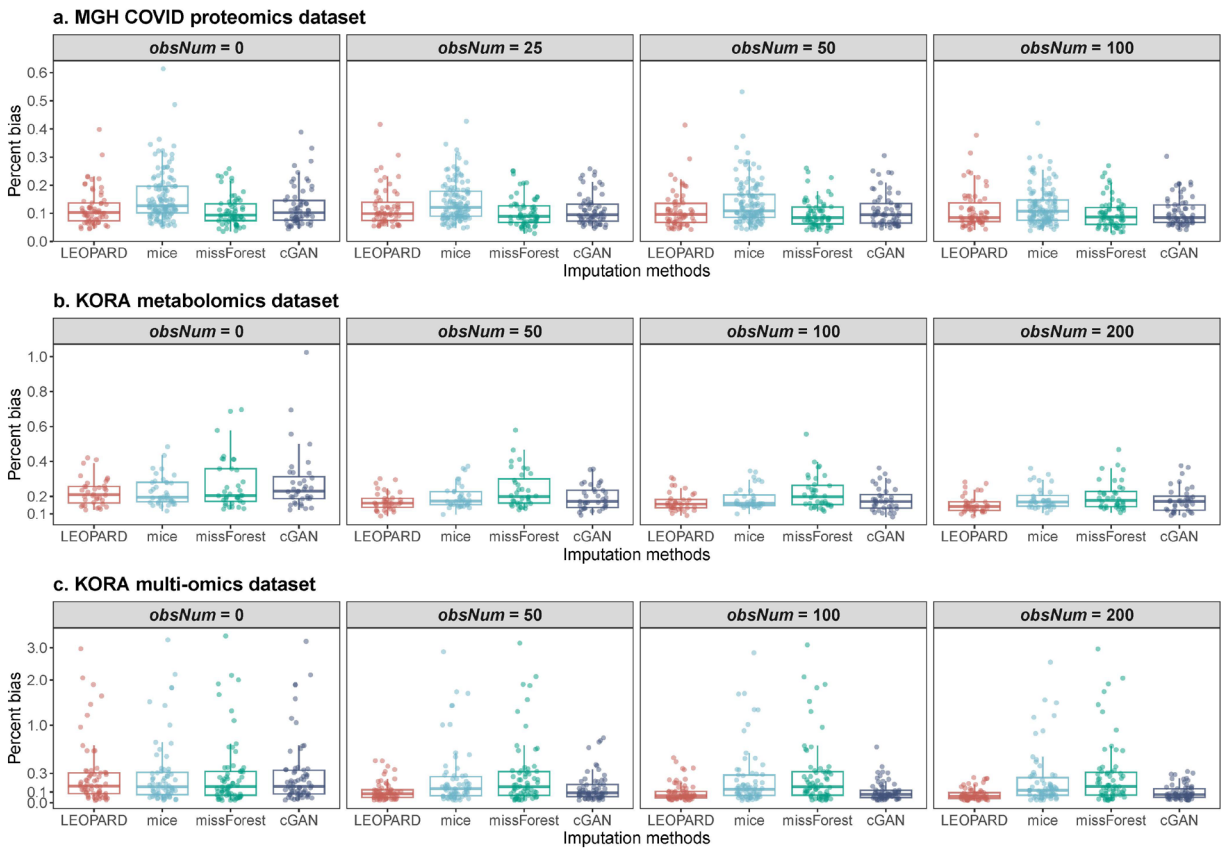


Fig. 3. PB of imputed results for the test sets of three benchmark datasets.

a-c, PB evaluated on $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$ of the MGH COVID proteomics dataset (**a**), KORA metabolomics dataset (**b**), and KORA multi-omics dataset (**c**). Each dot represents a PB value for a variable.

Evaluation on the mono-omics datasets For the MGH COVID proteomics dataset, missForest overall exhibits the lowest PB, whereas the neural network-based methods cGAN and LEOPARD show comparable performances. Specifically, when *obsNum* is 100, the UMAP representation (Fig. 4 (a)) reveals that the clusters of the imputed data generated by all four methods (green dots) closely approximated the actual data (blue dots), indicating high similarity between the imputed and original datasets.

Interestingly, we observe that missForest produces the most unstable result for the KORA metabolomics dataset, showing the largest interquartile range (IQR) of 0.186 when *obsNum* is 0. In comparison, LEOPARD achieves the smallest IQR of 0.094 under the same *obsNum*, while mice and cGAN obtain IQR values of 0.112 and 0.125, respectively. When *obsNum* increases to 200, all four methods—mice, missForest, cGAN, and LEOPARD—yield low median PB values of 0.168, 0.177, 0.172, and 0.142, respectively. However, the UMAP embeddings of $\widehat{\mathcal{D}}^{\text{test}}_{v=v2,t=t2}$ generated by missForest and cGAN are mixed with those of $\mathcal{D}^{\text{test}}_{v=v2,t=t1}$, instead of $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$, implying that they overfit to $t1$ and do not generalize well to the second timepoint (Fig. 4 (b)). Moreover, the UMAP embeddings of $\widehat{\mathcal{D}}^{\text{test}}_{v=v2,t=t2}$ from mice only partly overlap with those of $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$, suggesting that some variations in the observed data have not been captured. In contrast, the embeddings of the data imputed by LEOPARD widely spread within the embedding space of the observed data, which demonstrates that LEOPARD has effectively learned and approximated the observed data's distribution.

Evaluation on the KORA multi-omics dataset In contrast to mono-omics datasets, where both views are from the same omics type, multi-omics datasets require imputation methods to capture more intricate relationships between omics data to ensure accurate results. In our evaluation of the KORA multi-omics data, all methods show some extremely high PB values when *obsNum* is 0, though LEOPARD slightly outperforms with a median PB of 0.152. When *obsNum* increases to 200, LEOPARD lowers its median PB to 0.061, while its closest competitor, cGAN, reduces its median PB from 0.158 to 0.076. In contrast, the performances of mice (from 0.153 to 0.117) and missForest (from 0.156 to 0.159) do not show similar improvements. The UMAP visualization (Fig. 4 (c)) further reveals a limited ability of mice and missForest to capture signals from the $t2$ timepoint, as their embeddings of $\widehat{\mathcal{D}}^{\text{test}}_{v=v2,t=t2}$ cluster with $\mathcal{D}^{\text{test}}_{v=v2,t=t1}$, not $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$. LEOPARD's performance is further validated by a high similarity between the distributions of the imputed and observed data embeddings in the UMAP space.

Analysis on imputed data with extremely high PB values We further investigate the extremely high PB values (> 0.8) observed in the KORA multi-omics dataset. We notice that proteins with low abundance (< 4.0) tend to exhibit extremely high PB (> 0.8) in the imputed values obtained under *obsNum* = 0 (Extended Fig. 1). For instance, SCF (stem cell factor), with a median abundance of 9.950, has a PB of 0.090 calculated from the LEOPARD-imputed data. In contrast, NT3 (Neurotrophin-3), with a much lower median abundance of 0.982, shows a PB of 1.187 calculated from the same imputed data. Our experiments show that these extremely high PB values can be substantially lowered by increasing *obsNum* (Figure 3, Extended Fig. 1).

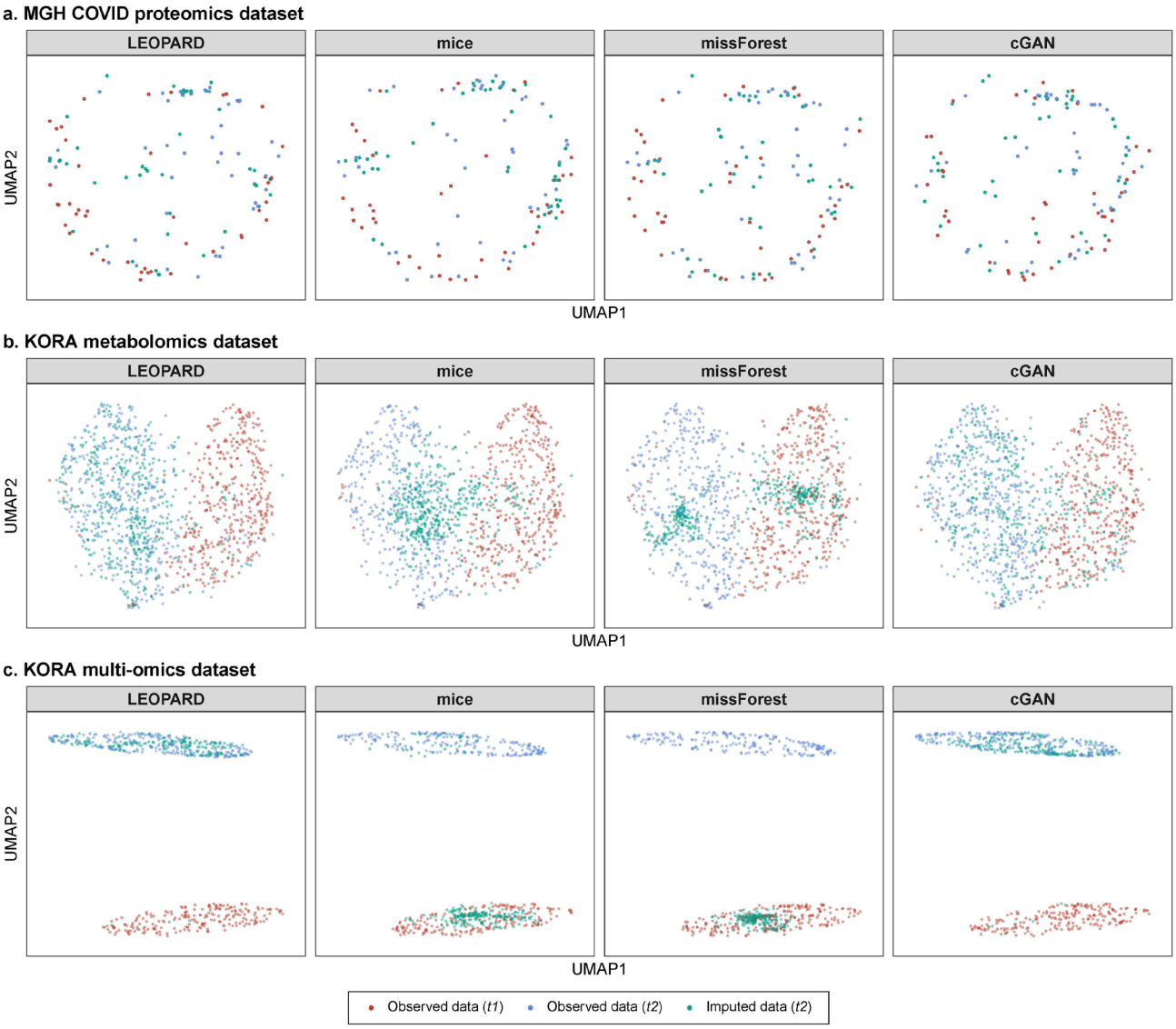


Fig. 4. UMAP representations of the imputed values and corresponding observed data from the benchmark datasets.

a-c, UMAP models are initially fitted with the training data from the MGH COVID proteomics dataset (**a**, $t1$: D0, $t2$: D3), KORA metabolomics dataset (**b**, $t1$: F4, $t2$: FF4), and KORA multi-omics dataset (**c**, $t1$: S4, $t2$: F4). Subsequently, the trained models are applied to the corresponding observed data (represented by red and blue dots for $t1$ and $t2$) and the data imputed by different methods (represented by green dots) under the setting of *obsNum* = 100 for the MGH COVID dataset and *obsNum* = 200 for the two KORA-derived datasets. The distributions of red and blue dots illustrate the variation across the two timepoints, while the similarity between the distributions of blue and green dots indicates the quality of the imputed data. A high degree of similarity suggests a strong resemblance between the imputed and observed data.

Case studies

We further conduct two case studies to investigate whether biological signals are preserved in the imputed data from different methods, with *obsNum* = 0.

Regression analysis of age-associated metabolites We use linear regression models to identify metabolites associated with age, controlling for sex. The models are separately fitted using the observed data and different imputed data that correspond to 36 metabolites from $\mathcal{D}^{test}_{v=v2,t=t2}$ (N = 417) of the KORA metabolomics dataset. Of the 18 metabolites significantly associated with age after Bonferroni correcting for multiple testing ($P < 0.05/36$, Methods) in the observed data, 17 are also significant in the data imputed by LEOPARD (see Fig. 5 (a)). In contrast, only one metabolite is significantly associated with age in the data imputed by mice or missForest. No metabolite is identified to be significant in the data imputed by cGAN.

Classification analysis of chronic kidney disease We use balanced random forest[40] classifiers to predict chronic kidney disease (CKD). The classifiers are fitted individually using the observed and different imputed data that correspond to 66 proteins from $\mathcal{D}^{test}_{v=v2,t=t2}$ (N = 212) of the KORA multi-omics dataset. CKD cases are defined as having an estimated glomerular filtration rate (eGFR) < 60 mL/min/1.73m^2 [41]; 36 individuals in the dataset are identified as CKD cases. We train the classifiers using identical hyperparameters and use leave-one-out-cross-validation to evaluate the performance (Methods). We obtain an F1 Score of 0.544 in the model built with the observed data; the model built with the data imputed by LEOPARD achieves the closest performance with an F1 Score of 0.403 (Fig. 5 (b), Extended Table 1). LEOPARD outperforms its competitors with an area under the receiver operating characteristic curve (AUROC) of 0.725 and an area under the precision-recall curve (AUPRC) of 0.435 (Fig. 5 (c)).

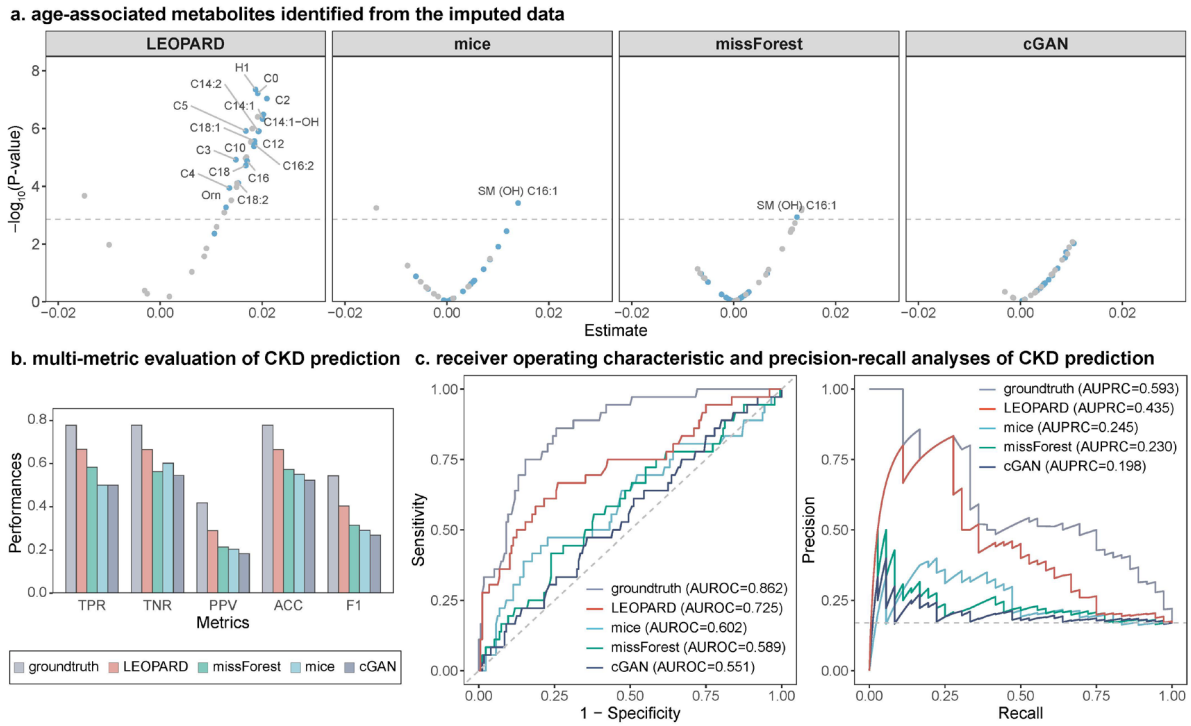


Fig. 5. Evaluation of the data imputed by different methods on regression and classification analyses.

a, Volcano plots display age-associated metabolites detected in the $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$ and $\widehat{\mathcal{D}}^{\text{test}}_{v=v2,t=t2}$ ($obsNum$ = 0) of the KORA metabolomics dataset (N = 417). 18 significant metabolites (P < 0.05/36) identified in the observed data are shown in blue. Replicated metabolites from the data imputed by different methods are marked with labels. **b-c**, CKD classification evaluated using $\mathcal{D}^{\text{test}}_{v=v2,t=t2}$ and $\widehat{\mathcal{D}}^{\text{test}}_{v=v2,t=t2}$ ($obsNum$ = 0) from the KORA multi-omics dataset (N = 212, $N_{positive}$ = 36, $N_{negative}$ = 176). Models are trained using the balanced random forest algorithm with identical hyperparameters and evaluated using leave-one-out-cross-validation. Evaluation metrics in the bar plot include true positive rate (TPR, also known as sensitivity), true negative rate (TNR, also known as specificity), positive predictive value (PPV, also known as precision), accuracy (ACC), and F1 score. The dashed lines in the ROC and PR curves represent the performance of a hypothetical model with no predictive capability.

Discussion

In this study, we introduce LEOPARD, a novel architecture for missing view completion designed for multi-timepoint omics data. The performance of LEOPARD is comprehensively assessed through simulations and case studies using three real human omics datasets. Additional interesting findings emerged from our evaluation.

As illustrated in the UMAP plots, the MGH COVID proteomics data (Fig. 4 (a)) from D0 ($t1$, in red) and D3 ($t2$, in blue) show relatively low variation, while the KORA metabolomics data (Fig. 4 (b)) from F4 ($t1$, in red) and FF4 ($t2$, in blue) exhibit more substantial variations, potentially due to biological variations spanning seven years and technical variation from different analytical kits in the KORA data. In the MGH COVID dataset, we observed that LEOPARD performs slightly worse than its competitors. This can be attributed to the inherent differences in the representation learning process of LEOPARD and its competitors. By directly learning mappings between views, the methods developed for cross-sectional data can exploit the input data to learn detailed, sample-specific patterns, while the representation learning in LEOPARD primarily focuses on more compact and generalized structures related to views and timepoints, potentially neglecting detailed information specific to individual samples. The MGH COVID dataset, having a high similarity between the data from D0 and D3, allows the cross-sectional imputation methods to effectively apply the mappings learned from one timepoint to another. As the data variation between two timepoints increases, LEOPARD's advantages become increasingly evident, while its competitors tend to overfit the training data and fail to generalize well to the new timepoint (Fig. 4 (b, c)).

For the KORA multi-omics dataset, we observed that the extremely high PB values are associated with low analyte abundances. For the same absolute error in imputed values, a variable with a low analyte abundance will have a higher absolute error ratio, leading to a higher PB than those variables with high abundances. Additionally, protein levels quantified using the Olink platform are represented as relative quantities. Data from different measurements can contain technical variations that arise from experimental factors, such as instrument performance, and normalization methods used for relative quantification. When *obsNum* is 0, LEOPARD is trained without any data from $v2$ at $t2$, and thus cannot account for the technical variations exclusive to that part. By incorporating a few observed samples from the second timepoint into the training process, the model can better capture the data distribution and technical variation of the missing part, which contributes to a substantial reduction in high PB values.

As data imputation inevitably incurs a loss of information, we conducted two case studies to assess the preservation of biological information in the imputed data. Despite all four imputation methods producing similar PB when *obsNum* is 0 (Fig. 3 (c)), the case studies showed that the data imputed by LEOPARD provided performances closest to the observed data, while the data imputed by mice, missForest, and cGAN showed a substantial loss of biological information (Fig. 5). This outcome highlights the importance of case studies for a reliable evaluation of imputed data. Our experiments are restricted by data availability to two timepoints, but in principle LEOPARD can accommodate additional timepoints and is well-suited for analyses involving multiple omics per timepoint. LEOPARD not only imputes missing data for downstream analyses, but also facilitates the exploration of temporal dynamics through its extracted temporal representations, providing an additional opportunity for advances in predictive healthcare with multi-timepoint omics data.

With advancements in omics measurement technology and the increasing availability of longitudinal data, missing data in multi-timepoint, multi-view data is becoming a prominent issue. Our study highlights the necessity for imputation methods in this context, and our method LEOPARD represents an early attempt to address this issue. We anticipate further developments in imputation methods that exhibit high generalization ability, robustness to low analyte abundance, and preservation of biological variations.

Methods

Terminology

The following outlines the terminology used throughout this study.

- Multi-view: several modalities collected in one cohort[2]. It is common for samples to be examined from multiple perspectives using different platforms or technologies, yielding different views of the data.
- Mapping: the transformation that a model learns to apply from an input space to an output space.
- Embeddings: low-dimensional representations of high-dimensional data.

Data preprocessing

MGH COVID dataset The proteomics data from the MGH COVID study were obtained using plasma samples and the Olink® Explore 1536 platform (Olink Proteomics, Watertown, MA, USA), which consists of 1,472 proteins across four categories: inflammation, oncology, cardiometabolic, and neurology[17]. The platform enables relative quantification of analyte concentrations in the form of $\log_2$-scaled normalized protein expression (NPX) values, where higher values correspond to higher protein levels. To facilitate comparison with the KORA multi-omics dataset, we selected proteins listed in the Olink® Target 96 Cardiometabolic and Inflammation assays to construct the MGH COVID proteomics dataset. Similar to the KORA multi-omics dataset, inflammation view was assumed to be incomplete. We select the proteins based on the following QC criteria: (1) no missing values; (2) at least 75% of measured sample values are equal to or above the limits of detection (LOD). After quality inspection (Table S1), a total of 79 and 66 proteins from the Cardiometabolic and Inflammation assays met all criteria. After removing one duplicated individual, proteomics data from 218 individuals are collected.

KORA datasets The metabolite profiling of the KORA S4 (March - April 2011), F4 (August 2008 - March 2009), and FF4 (February - October 2019) serum samples spans over a decade during which analytical procedures have been upgraded several times[26]. The targeted metabolomics data of F4 were measured with the analytical kit AbsoluteIDQ® p150[42,43], while S4 and FF4 data were quantified using the kit AbsoluteIDQ® p180[44–46] (Biocrates Life Sciences AG, Innsbruck, Austria). To assess the technical variation introduced by different kits, samples from 288 individuals from the F4 were remeasured (September-October 2019) using the p180 kit. Three manufacturer-provided quality control (QC) samples were added to each plate to

quantity the plate effect. Additionally, five external QC samples[47] were added to each plate when measuring using the p180 kit. The p150 and p180 kits allow simultaneous quantification of 163 and 188 metabolites, respectively. Only metabolites meeting the following five criteria were selected: (1) overlap between p150 and p180; (2) no missing or negative values; (3) at least 50% of measured sample values are equal to or above the LOD of corresponding plates; (4) median relative standard deviation (RSD) of QC samples < 25%; (5) Spearman correlation coefficients between the KORA F4 (remeasured) and F4 (original) > 0.5. In total, 106 and 104 metabolites satisfied all criteria for the KORA metabolomics and multi-omics datasets (Table S2). The metabolites for both the metabolomics and multi-omics datasets were further normalized using TIGER with its default setting to remove the plate effects. For the multi-omics dataset, we additionally used TIGER to remove the technical variation introduced by different kits following our previous protocol[26,48].

The proteomics data measurements of the KORA S4 and F4 were performed using plasma (S4, February 2020) and serum (F4, December 2016 - January 2017) samples with the Olink® Target 96 Inflammation panel (Olink Proteomics, Uppsala, Sweden), which covers 92 proteins[49]. In quality inspection, only proteins meeting the following criteria were selected: (1) no missing values; (2) at least 75% of measured sample values are equal to or above the LOD. After the inspection, 66 proteins were selected for the KORA multi-omics dataset (Table S3).

Training the cGAN model

Architecture implementation The cGAN[30] extends the original GAN[50] model by introducing additional information into the generation process, thereby providing more control over the generated output. In our context, the completion of missing views is conditioned on the data from the observed views. Moreover, we enhanced the baseline cGAN model with an auxiliary classifier[33] to ensure that the imputed view can be compatible with the corresponding observed view.

Specifically, the generator G_{cGAN} learns the complex mappings between $\mathcal{D}^{\text{train}}_{v=v1,t}$ and $\mathcal{D}^{\text{train}}_{v=v2,t}$, with $t = t1$ if the data of one view at $t2$ are completely missing. The reconstruction loss $\mathcal{L}_{Rec_cGAN}$ quantifies the differences between the actual and reconstructed data. The discriminator D_{cGAN} computes the adversarial loss $\mathcal{L}_{Adv_cGAN}$ by distinguishing if data are real $\mathcal{D}^{\text{train}}_{v=v2,t=t1}$ or generated $\widehat{\mathcal{D}}^{\text{train}}_{v=v2,t=t1}$. D_{cGAN} also has an auxiliary classifier that computes the

auxiliary loss $\mathcal{L}_{Aux_cGAN}$ by predicting whether the pairs of views are real, i.e.,$\left(\mathcal{D}^{\text{train}}_{v=v1,t=t1}, \mathcal{D}^{\text{train}}_{v=v2,t=t1}\right)$, or fake, i.e., $\left(\mathcal{D}^{\text{train}}_{v=v1,t=t1}, \widehat{\mathcal{D}}^{\text{train}}_{v=v2,t=t1}\right)$. The final loss is defined as:

$$\mathcal{L}_{cGAN} = w_{Rec_cGAN} \times \mathcal{L}_{Rec_cGAN} + w_{Adv_cGAN} \times \mathcal{L}_{Adv_cGAN} + w_{Aux_cGAN} \times \mathcal{L}_{Aux_cGAN},$$

where w_{Rec_cGAN}, w_{Adv_cGAN} and w_{Aux_cGAN} represent weights for each loss. After training, the generator G_{cGAN} is applied to $\mathcal{D}^{\text{test}}_{v=v1,t=t2}$ to generate $\widehat{\mathcal{D}}^{\text{test}}_{v=v2,t=t2}$.

The generator of the cGAN model consists of several residual blocks[51] and uses the parametric rectified linear unit (PReLU)[52] as its activation function. Each residual block includes batch normalization[53] as necessary. MSE loss serves as $\mathcal{L}_{Rec_cGAN}$. Both MSE loss and binary cross-entropy (BCE) loss are considered for $\mathcal{L}_{Adv_cGAN}$ and $\mathcal{L}_{Aux_cGAN}$, determined by the hyperparameter tuning experiments. MSE and BCE loss are defined as:

$$\mathcal{L}_{MSE} = \frac{1}{N}\left\|x^i - \hat{x}^i\right\|_2^2,$$

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N}\left[y^i \log\left(\hat{y}^i\right) + \left(1 - y^i\right)\log\left(1 - \hat{y}^i\right)\right],$$

where N is the number of samples, x^i and $\hat{x}^i$represents the true and estimated values for the i-th sample, while y^i and $\hat{y}^i$ represents the true and predicted labels for the i-the sample.

The cGAN model is trained with the Adam optimizer[54], with a mini-batch size of 16 for the MGH COVID dataset and 32 for the two KORA-derived datasets. The model is implemented using PyTorch[55] (v1.11.0), PyTorch Lightning[56] (v1.6.4), and tensorboard[57] (v2.10.0), and run on a graphics processing unit (GPU) operating compute unified device architecture (CUDA, v11.3.1).

Hyperparameter optimization We utilized the training (64%) and validation (16%) sets to optimize the hyperparameters of the cGAN model. We employed a grid search over various combinations of hidden layer sizes, numbers of residual blocks, batch normalization settings, and weights for different losses. We determined the number of training epochs based on early stopping triggered by the MSE reconstruction accuracy calculated on the observed data from the validation sets.

Our experiments included variations in the number of hidden neurons for both the generator and discriminator, with options including 32, 64, 128, and 256. The number of residual blocks spanned from 2 to 6. We applied a weight of 0.5 to $\mathcal{L}_{Rec_cGAN}$ and a weight of 0.25 to $\mathcal{L}_{Adv_cGAN}$ and $\mathcal{L}_{Aux_cGAN}$ for all our performance evaluations.

Training the LEOPARD

Architecture implementation View-specific pre-layers E_{pre}^{v} are used to embed input data $x_{v,t}^{i} \in \mathcal{D}_{v,t}^{\text{train}}$ of different views into dimensionally uniform embeddings $z_pre_{v,t}^{i}$. Here, i represents the data from the i-th individual. The representation disentangler of LEOPARD comprises a content encoder E_c and a temporal encoder E_t, both shared by input data across different views and timepoints. This module learns a timepoint-invariant content feature $z_content_{v,t}^{i}$ and temporal feature $z_temporal_{v,t}^{i}$ from the embeddings of input sample $z_pre_{v,t}^{i}$. Following the encoding process, the generator G employs the AdaIN technique to re-entangle content representation and temporal knowledge:

$$\text{AdaIN}(z_content^{i}, z_temporal^{i}) = \sigma(z_temporal^{i}) \times \frac{z_content^{i} - \mu(z_content^{i})}{\sigma(z_content^{i})} + \mu(z_temporal^{i}),$$

where μ and σ denote the mean and standard deviation operations respectively. View-specific post-layers E_{post}^{v} convert the re-entangled embeddings back to omics data $\hat{x}_{v,t}^{i}$. Our discriminator D is trained to classify whether an input is a real sample or a generated output coming from G and E_{post}^{v}. D produces the same number of outputs as the source classes of the observed data, each corresponding to the data of one view at one timepoint. For a sample belonging to source class $c_{v,t}$, we penalize D during the update cycle of D if its output incorrectly classifies a real data instance as false or a generated data instance as true for $c_{v,t}$; when updating G, we only penalize G if D correctly identifies the generated data instance as false for $c_{v,t}$.

In our study, we defined the contrastive loss $\mathcal{L}_{Con}$ as the mean of the NT-Xent losses calculated separately for content and temporal representations. The NT-Xent loss is formulated as follows:

$$\mathcal{L}_{NT-Xent}(z^{i}, z^{j}) = -\log \frac{\exp(\text{sim}(z^{i}, z^{j})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z^{i}, z^{k})/\tau)},$$

where z^{i} and z^{j} are embeddings of a positive pair (i, j), τ is a temperature factor that scales the similarities, and $\mathbb{1}_{[k \neq i]}$ is an indicator function that equals 1 when $k \neq i$ and 0 otherwise. And $\text{sim}(\cdot)$ denotes cosine similarity, defined as:

$$\text{sim}(\text{a, b}) = \frac{\text{a} \cdot \text{b}}{\|a\|_2 \, \|b\|_2}.$$

In each training iteration, we generate data for the missing view before calculating the loss. This allows the generated data to be factorized into corresponding content and temporal

representations, further facilitating loss minimization. For the $\mathcal{L}_{NT-Xent}$ calculated on content representations, positive pairs are defined as $(z_content^{i}_{v=v1,t=t1}, z_content^{i}_{v=v1,t=t2})$ and $(z_content^{i}_{v=v2,t=t1}, z_\widehat{content}^{i}_{v=v2,t=t2})$, which are the same kind of content embeddings from the same individuals across different timepoints. Similarly, the positive pairs of temporal representations are $(z_temporal^{i}_{v=v1,t=t1}, z_temporal^{i}_{v=v2,t=t1})$ and $(z_temporal^{i}_{v=v1,t=t2}, z_\widehat{temporal}^{i}_{v=v2,t=t2})$. The representation loss $\mathcal{L}_{Rep}$ is the mean of the MSE losses calculated for content and temporal representations. For each type of representation, LEOPARD measures the MSE between this specific representation factorized from the actual data and the same kind of representation factorized from the reconstructed data. The reconstruction loss $\mathcal{L}_{Rec}$ quantifies the discrepancies between the actual and reconstructed data. The generator can arbitrarily produce data for any source classes given the content and temporal representations. To ensure the representation disentangler can capture the highly structured data pattern, we only compute the $\mathcal{L}_{Rec}$ on the data generated from content and temporal representations derived from different source classes. For instance, $(z_content^{i}_{v=v1,t=t1}, z_temporal^{i}_{v=v2,t=t1})$ or $(z_\widehat{content}^{i}_{v=v2,t=t2}, z_temporal^{i}_{v=v1,t=t1})$. We avoid evaluating data generated from representation pairs of the same views and timepoints, such as $(z_content^{i}_{v=v1,t=t2}, z_temporal^{i}_{v=v1,t=t2})$. Similar to the cGAN model described in the previous section, the adversarial loss $\mathcal{L}_{Adv}$ is also computed based on MSE. The final loss of LEOPARD is defined as:

$$\mathcal{L}_{LEOPARD} = w_{Con} \times \mathcal{L}_{Con} + w_{Rep} \times \mathcal{L}_{Rep} + w_{Rec} \times \mathcal{L}_{Rec} + w_{Adv} \times \mathcal{L}_{Adv},$$

where w_{Con}, w_{Rep}, w_{Rec} and w_{Adv} are the weights of the losses.

The encoders, generator, and discriminator of LEOPARD are built from blocks of layers without skip connections. Each block starts with a dense layer. An instance normalization layer[58] is added after the dense layer for the content encoder. The encoders and generator use the PReLU as its activation function, while the discriminator uses the sigmoid function. A dropout layer[59] is incorporated after the activation layer, where necessary.

The LEOPARD model is trained with the Adam optimizer, with a mini-batch size of 16 for the MGH COVID dataset and 64 for the two KORA-derived datasets. The model is implemented under the same computational environment as the cGAN model.

Hyperparameter optimization The training (64%) and validation (16%) sets were used to optimize hyperparameters, aiming to effectively capture data structure for existing data

reconstruction and achieve robust generalization for missing data imputation. We conducted a grid search across various combinations of hidden layer sizes, hidden layer numbers, dropout rates, loss weights, and temperatures for contrastive loss, focusing on improvement in reconstruction accuracy and reduction in contrastive loss computed on validation set.

The number of hidden neurons for the encoders, generator, and discriminator varied across 32, 64, 128, and 256, with the number of layers (inclusive of the pre-layer or post-layer) ranging from 2 to 5. We evaluated dropout rates of 0%, 30%, and 50%, as well as loss weights of 0.1, 0.5, 1, and contrastive loss temperatures of 0.01, 0.05, 1, 5, 10, and 100. The number of training epochs was determined by the model's saturation point in learning, which is when the MSE reconstruction loss ceased to decrease significantly. Our findings showed that higher numbers of hidden neurons and layers tended to yield worse performance on the validation set. Dropout was not used in the final model. The size of the post-layers is dataset-specific, determined by the variable numbers of different datasets.

Representation disentanglement

The disentanglement of content and temporal representations was evaluated using the KORA multi-omics dataset. LEOPARD was trained for 600 epochs, for each of which the disentanglement progress was visualized with the following steps: First, content and temporal representations were factorized from the metabolomics (S4 and F4) and proteomics data (S4). Then the generator imputed the proteomics data (F4) by incorporating the temporal information from the metabolomics data (F4) into the content representation from the proteomics data (S4). The generated proteomic data (F4) were then fed to the content and temporal encoders to extract the corresponding representations. Subsequently, these content or temporal representations of both the observed and imputed data were standardized to ensure all latent variables had a mean of zero and a standard deviation of one. Afterward, two separate UMAP models were built using the R package umap[60], each fitted to the content and temporal representations, with a configuration of *n_neighbors* = 15 and *min_dist* = 0.1. Lastly, scatter plots were generated using the R packages ggplot2[61] and ggsci[62]. Each point in the plot represents an individual sample, and the color indicates the data sources. The visualization epochs were selected experimentally to illustrate the progress of representation disentanglement during the training process.

Performance evaluation

The input data for methods mice, missForest, and cGAN include all training data from the first timepoint ($D^{\text{train}}_{v=v1,t=t1}$ and $D^{\text{train}}_{v=v2,t=t1}$) and training data of different *obsNum* from the second timepoint ($D^{\text{train}}_{v=v1,t=t2}$ and $D^{\text{train}}_{v=v2,t=t2}$). As LEOPARD can process observations where only one view is present, its input data include all the data from $D^{\text{train}}_{v=v1,t=t1}$, $D^{\text{train}}_{v=v2,t=t1}$, and $D^{\text{train}}_{v=v1,t=t2}$, and the data of different *obsNum* from $D^{\text{train}}_{v=v2,t=t2}$. When *obsNum* = 0, data of $v2$ at $t2$ are assumed as completely missing. The performances of mice, missForest, cGAN, and LEOPARD were evaluated on $D^{\text{test}}_{v=v2,t=t2}$ of three benchmark datasets.

For mice, imputed values were obtained from a single imputation (m = 1) using the default predictive mean matching (PMM)[63] algorithm, with the maximum number of iterations (*maxit*) set to five. For missForest, the imputation was performed using a 100-tree random forest[64] model, with the *maxit* value set to 10. The cGAN and LEOPARD models were trained using the hyperparameters previously described.

PB was selected as the performance metric as it quantifies the relative deviation of imputed values from actual observations, offering a more straightforward interpretation compared to metrics like RMSE and MAE. PB was calculated for each variable using the formula:

$$\text{PB} = \text{median}\left(\frac{\left|\hat{x}^i - x^i\right|}{x^i}\right).$$

PB results for each imputation method were visualized using dot and box plots, with each dot representing a variable in the specific dataset.

For the evaluation using UMAP, we first fitted a UMAP model using the data of $D^{\text{train}}_{v=v2,t=t1}$ and $D^{\text{train}}_{v=v2,t=t2}$. We then used the fitted model to embed the data of $D^{\text{test}}_{v=v2,t=t1}$, $D^{\text{test}}_{v=v2,t=t2}$, and $\widehat{D}^{\text{test}}_{v=v2,t=t2}$, where $\widehat{D}^{\text{test}}_{v=v2,t=t2}$ represents the imputed data for $D^{\text{test}}_{v=v2,t=t2}$ produced by different imputation methods. An imputation method is considered effective if the distribution of $\widehat{D}^{\text{test}}_{v=v2,t=t2}$ embeddings is highly similar to that of the $D^{\text{test}}_{v=v2,t=t2}$ embeddings. The UMAP models were fitted with the identical configurations described in the prior section.

Regression analysis of age-associated metabolites

We used observed and imputed test set data from the KORA metabolomics dataset for the identification of age-associated metabolites. The imputed data of mice, missForest, cGAN, and LEOPARD were obtained under the setting of *obsNum* = 0. The test set comprised the concentrations of 36 metabolites from 417 individuals.

For each of the observed or imputed data, we employed multivariate linear regression models, using the concentration of each metabolite as the response variable and age as the predictor variable, while controlling for sex. To mitigate the risk of false positives in multiple testing, we applied a Bonferroni correction to adjust the *P*-value significance threshold. The adjusted significance level was used to identify the metabolites significantly associated with age.

Classification analysis of chronic kidney disease

We used observed and imputed test set data from the KORA multi-omics dataset for CKD prediction. The data imputed by mice, missForest, cGAN, and LEOPARD were obtained under the setting of *obsNum* = 0. The test set consisted of NPX values of 66 proteins from 212 individuals, among which 36 were labeled as CKD (positive) cases. CKD cases were determined based on their eGFR values, which were computed from serum creatinine, sex, race, and age, using the Chronic Kidney Disease Epidemiology Collaboration (CKD-EPI) equation[65].

For each type of raw or imputed data, we trained a BRF model using the Python library imbalanced-learn[66]. This model was specifically selected to address the dataset imbalance and reduce the risk of overfitting to the majority class. All models were trained with default hyperparameters (*criterion* = "gini", *min_samples_split* = 2, *min_samples_leaf* = 1, *max_features* = "sqrt", *bootstrap* = True), except for *n_estimators* = 1000 and *class_weight* = "balanced_subsample". Due to the limited sample size, we validated the performance using the leave-one-out-cross-validation strategy, allowing maximal use of data for both training and validation. Performance metrics were calculated using the R package caret[67]. These metrics provided a comprehensive understanding of the predictive power of the observed and imputed data.

The ROC curves were plotted to illustrate the trade-off between sensitivity and 1-specificity at varying decision thresholds. Considering the imbalance in our dataset, and with our primary interest in the positive class, which is also the minority, we further plotted PR curves to depict the trade-off between precision and recall at different thresholds for the classifiers trained with the different data. For PR curves, the baseline performance of a non-discriminative model was determined by the proportion of positive cases, which was 36/212 = 0.170 in our study. Both the ROC and PR curves were plotted using the R package precrec[68].

Data availability

The MGH COVID data, published by the original authors, have been deposited to Mendeley Data (http://dx.doi.org/10.17632/nf853r8xsj). The KORA data are governed by the General Data Protection Regulation (GDPR) and national data protection laws, with additional restrictions imposed by the Ethics Committee of the Bavarian Chamber of Physicians to ensure data privacy of the study participants. Therefore, the data cannot be made freely available in a public repository. However, researchers with a legitimate interest in accessing the data may submit a request through an individual project agreement with KORA via the online portal (https://www.helmholtz-munich.de/en/epi/cohort/kora). Upon receipt of the request, the data access committee will review the application and, subject to approval, provide the researcher with a data usage agreement. The specific variable names and sample IDs for each dataset, determined following the QC process, can be found at: https://github.com/HAN-Siyu/LEOPARD.

Code availability

The source code and implementation details of LEOPARD are freely available at GitHub repository (https://github.com/HAN-Siyu/LEOPARD).

References

1. Vasaikar, S. V *et al.* A comprehensive platform for analyzing longitudinal multi-omics data. *Nat. Commun.* **14**, 1684 (2023).

2. Avants, B. B., Tustison, N. J. & Stone, J. R. Similarity-driven multi-view embeddings from high-dimensional biomedical data. *Nat. Comput. Sci.* **1**, 143–152 (2021).

3. Vandereyken, K., Sifrim, A., Thienpont, B. & Voet, T. Methods and applications for single-cell and spatial multi-omics. *Nat. Rev. Genet.* 1–22 (2023) doi:10.1038/s41576-023-00580-2.

4. Mitra, R. *et al.* Learning from data with structured missingness. *Nat. Mach. Intell.* **5**, 13–23 (2023).

5. Flores, J. E. *et al.* Missing data in multi-omics integration: Recent advances through artificial intelligence. *Front. Artif. Intell.* **6**, 1098308 (2023).

6. Zhang, L., Zhao, Y., Zhu, Z., Shen, D. & Ji, S. Multi-View Missing Data Completion. *IEEE Trans. Knowl. Data Eng.* **30**, 1296–1309 (2018).

7. Rosato, R., Pagano, E., Testa, S., Zola, P. & di Cuonzo, D. Missing data in longitudinal studies: Comparison of multiple imputation methods in a real clinical setting. *J. Eval. Clin.*

Pract. **27**, 34–41 (2021).

8. Buuren, S. van & Groothuis-Oudshoorn, K. **mice** : Multivariate Imputation by Chained Equations in *R. J. Stat. Softw.* **45**, (2011).

9. Stekhoven, D. J. & Bühlmann, P. MissForest--non-parametric missing value imputation for mixed-type data. *Bioinformatics* **28**, 112–8 (2012).

10. Troyanskaya, O. *et al.* Missing value estimation methods for DNA microarrays. *Bioinformatics* **17**, 520–5 (2001).

11. Samad, M. D. & Yin, L. Non-linear regression models for imputing longitudinal missing data. in *2019 IEEE International Conference on Healthcare Informatics (ICHI)* 1–3 (IEEE, 2019). doi:10.1109/ICHI.2019.8904528.

12. Tran, L., Yin, X. & Liu, X. Disentangled Representation Learning GAN for Pose-Invariant Face Recognition. in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1283–1292 (IEEE, 2017). doi:10.1109/CVPR.2017.141.

13. Huang, X. & Belongie, S. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. in *2017 IEEE International Conference on Computer Vision (ICCV)* 1510–1519 (IEEE, 2017). doi:10.1109/ICCV.2017.167.

14. Yang, F.-E., Chang, J.-C., Tsai, C.-C. & Wang, Y.-C. F. A Multi-Domain and Multi-Modal Representation Disentangler for Cross-Domain Image Manipulation and Classification. *IEEE Trans. Image Process.* **29**, 2795–2807 (2020).

15. Choi, Y., Uh, Y., Yoo, J. & Ha, J.-W. StarGAN v2: Diverse Image Synthesis for Multiple Domains. in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 8185–8194 (IEEE, 2020). doi:10.1109/CVPR42600.2020.00821.

16. Chou, J. & Lee, H.-Y. One-Shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization. in *Interspeech 2019* 664–668 (ISCA, 2019). doi:10.21437/Interspeech.2019-2663.

17. Filbin, M. R. *et al.* Longitudinal proteomic analysis of severe COVID-19 reveals survival-associated signatures, tissue-specific cell death, and cell-cell interactions. *Cell reports. Med.* **2**, 100287 (2021).

18. Holle, R., Happich, M., Löwel, H. & Wichmann, H. E. KORA - A research platform for

population based health research. *Gesundheitswesen* **67**, (2005).

19. Yao, Y. *et al.* Longitudinal associations between metabolites and long-term exposure to ambient air pollution: Results from the KORA cohort study. *Environ. Int.* **170**, 107632 (2022).

20. Herder, C. *et al.* Environmental risk factors of incident distal sensorimotor polyneuropathy: Results from the prospective population-based KORA F4/FF4 study. *Sci. Total Environ.* **858**, 159878 (2023).

21. Yao, Y. *et al.* Longitudinal associations between metabolites and immediate, short- and medium-term exposure to ambient air pollution: Results from the KORA cohort study. *Sci. Total Environ.* 165780 (2023) doi:10.1016/j.scitotenv.2023.165780.

22. van Buuren, S. *Flexible Imputation of Missing Data, Second Edition*. (Chapman and Hall/CRC, 2018). doi:10.1201/9780429492259.

23. Wang-Sattler, R. *et al.* Novel biomarkers for pre-diabetes identified by metabolomics. *Mol. Syst. Biol.* **8**, 615 (2012).

24. Huang, J. *et al.* Machine learning approaches reveal metabolic signatures of incident chronic kidney disease in individuals with prediabetes and type 2 diabetes. *Diabetes* **69**, 2756–2765 (2020).

25. Huang, J. *et al.* Validation of Candidate Phospholipid Biomarkers of Chronic Kidney Disease in Hyperglycemic Individuals and Their Organ-Specific Exploration in Leptin Receptor-Deficient db/db Mouse. *Metabolites* **11**, (2021).

26. Han, S. *et al.* TIGER: technical variation elimination for metabolomics data using ensemble learning architecture. *Brief. Bioinform.* **23**, (2022).

27. Yang, M. *et al.* Partially View-aligned Representation Learning with Noise-robust Contrastive Loss. in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 1134–1143 (IEEE, 2021). doi:10.1109/CVPR46437.2021.00119.

28. Wen, J. *et al.* Unified Embedding Alignment with Missing Views Inferring for Incomplete Multi-View Clustering. *Proc. AAAI Conf. Artif. Intell.* **33**, 5393–5400 (2019).

29. Ma, M. *et al.* SMIL: Multimodal Learning with Severely Missing Modality. *Proc. AAAI Conf. Artif. Intell.* **35**, 2302–2310 (2021).

30. Mirza, M. & Osindero, S. Conditional Generative Adversarial Nets. (2014).

31. Shang, C. *et al.* VIGAN: Missing view imputation with generative adversarial networks. in *2017 IEEE International Conference on Big Data (Big Data)* 766–775 (IEEE, 2017). doi:10.1109/BigData.2017.8257992.

32. Cai, L., Wang, Z., Gao, H., Shen, D. & Ji, S. Deep Adversarial Learning for Multi-Modality Missing Data Completion. in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* 1158–1166 (ACM, 2018). doi:10.1145/3219819.3219963.

33. Odena, A., Olah, C. & Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. in *Proceedings of the 34th International Conference on Machine Learning - Volume 70* 2642–2651 (JMLR.org, 2017). doi:10.5555/3305890.3305954.

34. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. in *Proceedings of the 37th International Conference on Machine Learning* (JMLR.org, 2020). doi:10.48550/arXiv.2002.05709.

35. Mathieu, M., Couprie, C. & LeCun, Y. Deep multi-scale video prediction beyond mean square error. (2015).

36. Zhao, H., Gallo, O., Frosio, I. & Kautz, J. Loss Functions for Image Restoration With Neural Networks. *IEEE Trans. Comput. Imaging* **3**, 47–57 (2017).

37. Mustafa, A., Mikhailiuk, A., Iliescu, D. A., Babbar, V. & Mantiuk, R. K. Training a Task-Specific Image Reconstruction Loss. in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* 21–30 (IEEE, 2022). doi:10.1109/WACV51458.2022.00010.

38. Liu, M.-Y. *et al.* Few-Shot Unsupervised Image-to-Image Translation. (2019).

39. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. (2018).

40. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

41. Nano, J. *et al.* Novel biomarkers of inflammation, kidney function and chronic kidney disease in the general population. *Nephrol. Dial. Transplant.* **37**, 1916–1926 (2022).

42. Römisch-Margl, W. *et al.* Procedure for tissue sample preparation and metabolite extraction for high-throughput targeted metabolomics. *Metabolomics* **8**, 133–142 (2012).

43. Illig, T. *et al.* A genome-wide perspective of genetic variation in human metabolism. *Nat. Genet.* **42**, 137–141 (2010).

44. Goek, O.-N. *et al.* Metabolites associate with kidney function decline and incident chronic kidney disease in the general population. *Nephrol. Dial. Transplant.* **28**, 2131–2138 (2013).

45. Dong, Q. *et al.* Metabolic Signatures Elucidate the Effect of Body Mass Index on Type 2 Diabetes. *Metabolites* **13**, 227 (2023).

46. Zukunft, S. *et al.* High-throughput extraction and quantification method for targeted metabolomics in murine tissues. *Metabolomics* **14**, 18 (2018).

47. Haid, M. *et al.* Long-Term Stability of Human Plasma Metabolites during Storage at −80 °C. *J. Proteome Res.* **17**, 203–211 (2018).

48. Shi, M. *et al.* Identification of candidate metabolite biomarkers for metabolic syndrome and its five components in population-based human cohorts. *Cardiovasc. Diabetol.* **22**, 141 (2023).

49. Herder, C. *et al.* A Systemic Inflammatory Signature Reflecting Cross Talk Between Innate and Adaptive Immunity Is Associated With Incident Polyneuropathy: KORA F4/FF4 Study. *Diabetes* **67**, 2434–2442 (2018).

50. Goodfellow, I. *et al.* Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2020).

51. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 770–778 (IEEE, 2016). doi:10.1109/CVPR.2016.90.

52. He, K., Zhang, X., Ren, S. & Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. in *2015 IEEE International Conference on Computer Vision (ICCV)* 1026–1034 (IEEE, 2015). doi:10.1109/ICCV.2015.123.

53. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37* 448–456 (JMLR.org, 2015). doi:10.5555/3045118.3045167.

54. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. (2014).

55. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. (2019).

56. Falcon, W., Borovec, J. & Eggert, N. PyTorchLightning/pytorch-lightning: Bug fixes, new docs. (2020) doi:10.5281/zenodo.3620922.

57. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. (2016).

58. Ulyanov, D., Vedaldi, A. & Lempitsky, V. Instance Normalization: The Missing Ingredient for Fast Stylization. (2016).

59. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).

60. Konopka, T. umap: Uniform Manifold Approximation and Projection. (2023).

61. Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*. (Springer-Verlag New York, 2016).

62. Xiao, N. ggsci: Scientific Journal and Sci-Fi Themed Color Palettes for 'ggplot2'. (2023).

63. Little, R. J. A. Missing-Data Adjustments in Large Surveys. *J. Bus. Econ. Stat.* **6**, 287–296 (1988).

64. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).

65. Levey, A. S. *et al.* A new equation to estimate glomerular filtration rate. *Ann. Intern. Med.* **150**, 604–12 (2009).

66. Lemaître, G., Nogueira, F. & Aridas, C. K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.* **18**, 1–5 (2017).

67. Kuhn, M. Building Predictive Models in R Using the caret Package. *J. Stat. Softw.* **28**, (2008).

68. Saito, T. & Rehmsmeier, M. Precrec: fast and accurate precision-recall and ROC curve calculations in R. *Bioinformatics* **33**, 145–147 (2017).

Acknowledgements

This project has received funding from the Innovative Medicines Initiative 2 Joint Undertaking (JU) under grant agreement No 821508 (CARDIATEAM). The JU receives support from the European Union's Horizon 2020 research and innovation programme and the European Federation of Pharmaceutical Industries and Associations (EFPIA).

The German Diabetes Center is supported by the German Federal Ministry of Health (Berlin, Germany) and the Ministry of Science and Culture in North-Rhine Westphalia (Düsseldorf, Germany). This study was supported in part by a grant from the German Federal Ministry of Education and Research to the German Center for Diabetes Research (DZD).

The KORA study was initiated and financed by the Helmholtz Zentrum München – German Research Center for Environmental Health, which is funded by the German Federal Ministry of Education and Research (BMBF) and by the State of Bavaria. Data collection in the KORA study is done in cooperation with the University Hospital of Augsburg.

We express our appreciation to all KORA study participants for their blood donation and time. We thank all participants for their long-term commitment to the KORA study, the staff for data collection and research data management and the members of the KORA Study Group (https://www.helmholtz-munich.de/en/epi/cohort/kora) who are responsible for the design and conduct of the KORA study.

We also extend our gratitude to all participants of the MGH COVID study, as well as the dedicated staff who have participated in the study's design, data collection, and project management.

The authors are grateful to Prof. Dr. Barbara Thorand (Institute of Epidemiology, Helmholtz Zentrum München) for her contribution to the collection of KORA proteomics data and for her constructive suggestions to improve this study.

The authors thank Yunhsiu Tai, Ming Cheng, Ruoyu Wang, Yuan Guo, Linrui Fan for their supports and suggestions during the development of LEOPARD.

Author contributions

Siyu Han designed and carried out the analyses, implemented the algorithms, interpreted the result, and wrote the manuscript.

Shixiang Yu, and Ying Li assisted in the development of the method.

Mengya Shi, Makoto Harada, Jianhong Ge, Flora Sam, Giuseppe Matullo assisted in interpreting the results.

Jiesheng Lin, Cornelia Prehn, Agnes Petrera, Jerzy Adamski, Karsten Suhre, Christian Gieger, Stefanie M. Hauck, Christian Herder, Michael Roden, Annette Peters, and Rui Wang-Sattler performed data acquisition and preparation of the KORA cohort data.

Na Cai and Francesco Paolo Casale assisted in the analysis and revised the manuscript.

Rui Wang-Sattler supervised the analyses, interpreted the results, and revised the manuscript.

All authors reviewed the final manuscript.

Competing interests

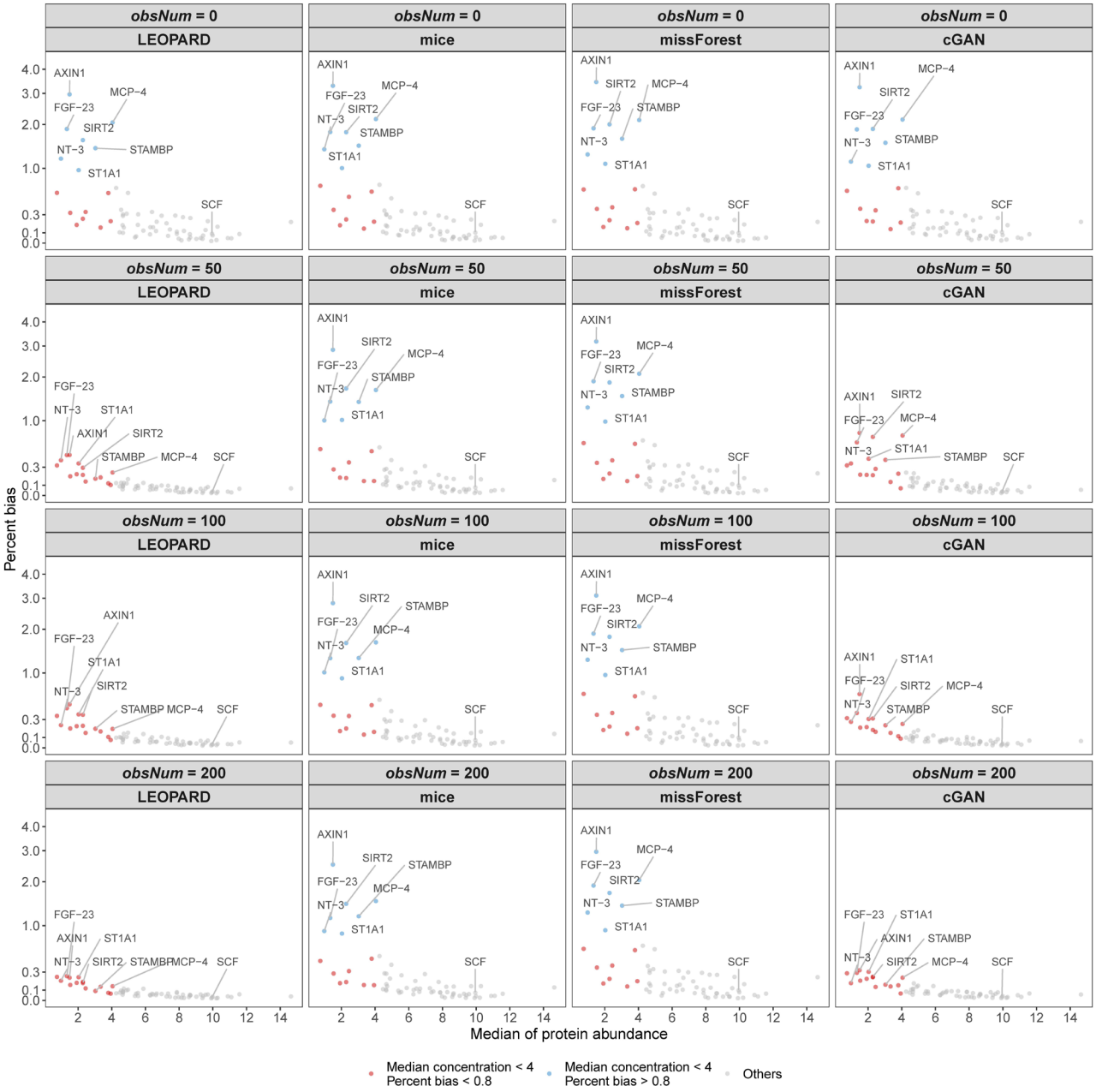
The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Rui Wang-Sattler.

Extended Table 1. Method performance of CKD prediction

Method	Sensitivity	Specificity	Precision	Balanced Accuracy	F1 Score	AUROC	PRRPC
raw	0.778	0.778	0.418	0.778	0.544	0.862	0.593
mice	0.500	0.602	0.205	0.551	0.290	0.602	0.245
missForest	0.583	0.563	0.214	0.573	0.313	0.589	0.230
cGAN	0.500	0.545	0.184	0.523	0.269	0.551	0.198
LEOPARD	0.667	0.665	0.289	0.666	0.403	0.725	0.435



Extended Fig. 1. Proteins with low abundance tend to exhibit high PB in the imputed values.

The proteins with low abundance (< 4.0) tend to exhibit extremely high PB (> 0.8) in the imputed values obtained under *obsNum* = 0. These extremely high PB values can be lowered by increasing *obsNum*.