# Deep Cooperative Reconstruction with Security Constraints in multi-view environments

Denis Maurel*, Sylvain Lefebvre†* and Jérémie Sublime*
*ISEP, 10 rue de Vanves, 92130 Issy-Les-Moulineaux, France
Email: jeremie.sublime@isep.fr
†Toyota Motor Corporation, Tokyo, Japan
Email:slvn-lefebvre@toyota-tokyo.tech

***Abstract*—Nowadays, we can observe a multiplication of multi-view data in domains such as marketing, bank administration, survey analysis, or social networks: We are dealing with large data bases that share a fair amount of data representing the same individual with different features depending on the data base.**

**In this context, one can use Machine Learning methods to analyze this fragmented data across several heterogeneous sources (called views). Such analysis is subject to several difficulties: First, not all individual will be present and represented in all data sites and views. And second, this type of cross site analysis raises several ethical questions on privacy issues as no local site should have direct access to data from the other sources.**

**To solve these problems, we present a method called the Cooperative Reconstruction System which aims at reconstructing information missing in some views in a multi-view context using information available in the other views. Furthermore, our method considers privacy issues and therefore achieves said reconstruction without direct data transfer from one view to another.**

***Index Terms*—Data reconstruction, multi-view Learning, Unsupervised learning, Neural networks**

## I. Introduction

With the proliferation of multi-view data in various domains such as marketing, bank administration or even survey analysis, it is necessary to develop techniques that can use these various data sources to provide more precise insights and analyses on these data. Multi-view learning can be a solution to these issues. As a sub-field of Machine Learning, it focuses on training models using databases distributed among several independent (but communicating) views, or feature sets. However, in order to be successful, these approaches need to solve two problems: First, a problem inherent to multi-view data is that while there will be large subsets of common individual present in several views, it will be very rare to have individuals represented in all views, thus leading to a problem of missing data. Second, concerns about which data should –or more often shouldn't– be made available and shared are rising and it is necessary for such a learning system to avoid unnecessary data sharing.

For example, consider a remote sensing or image capture device. Such instrument often use several sensors for redundancy and information fusion, provide multiple signals or point of views of the same observed phenomenon [1]. Amid growing concerns over privacy issues, it is also useful to consider how detailed information about individuals could be hidden.

The solution presented in this paper addresses these two main challenges: how to transfer usable information in a local view without transferring the original external data ? and how to reconstruct more or less reliable information from different sources to get the final result ?

To solve these problems, we present a system called the Cooperative Reconstruction System. Our system aims at the inference of missing data in a multi-view context with individuals whose information are complete in some views and missing in others. Another advantage of our system architecture is that the system can be naturally trained using a distributed architecture.

After encoding the original data using Autoencoders [2] to respect the security issues, our system uses a fully connected multi-layer deep network (called Links in this article) to combine the information coming from external views in to the local feature set. A smart weighting step follows this reconstruction step, to account for the respective views bias. The weighting method is presented in this paper and called Masked Weighting Method. The goal of this method is to (1) combine the information from different views, (2) reduce the weight of views with information which could hinder the cooperative reconstruction process [3], [4], and (3) reduce the impact of missing data during the unsupervised learning process [5].

This paper is organized as follow: a summary of the works related to this paper is presented in Section II, the Collaborative Reconstruction System is defined in-depth in Section IV and the empirical results are presented in Section V and VI. Finally, a conclusion along with some perspectives are given in Section VII.

## II. Related Works

As far as we know, the problem of cooperative reconstruction of an individual with security constraint has never been analyzed in the literature. However, it is possible to compare our approach with 3 existing fields related to specific aspects of our proposed system, namely Deep Multi-view representation learning (DMVRL), Collaborative Clustering (CC) and Collaborative Filtering (CF). Some works related

to the security aspect are also presented at the end of this section.

### A. Deep Multi-view representation learning

Multi-view representation learning is concerned with learning representations or features in a setting in which we have access to multiple unlabeled views of the data for representation learning while only one view is available at test time [6]. As one can see, this is closely related to our problem.

Models proposed to tackle this problem include deep learning based solutions similar to the ones presented in this paper.

In [7], the authors propose a system that tries to reconstruct shared representations that are available from 2 views available at a given time. This system relies on a split-autoencoder architecture that tries to minimize the sum of the reconstruction errors.

Evolutions of the previously described architecture include correlated autoencoders [8] that consists of two autoencoders and optimize the combination of canonical correlation between learned bottleneck representations and the reconstruction errors of the autoencoders. This architecture is used in [9] to learn vectorial word representations using parallel corpuses from two languages.

Canonical correlation analysis (CCA) in general has also been the subject of many architectures, deep or not, applied to tackle the problem of multi-view representation learning [8], [10]–[12].

A quick survey of the literature mentioned before shows that many algorithms are limited to only two views at a time, or make strong assumption about similitude and shared elements between the views, or more importantly have no concern for information having to be shared between views to train their models. It so happens that most practical applications have more than 2 views available at a time and do not allow for information to be shared or for an algorithm to access several views at the same time.

### B. Collaborative Clustering

The aim of Collaborative Clustering is to make different algorithms cooperate in order to find a consensus between the clusterings of several independent views. This is achieved through the exchange of information between the views. Systems built with this paradigm are similar to our reconstruction system in that they make independent views collaborate to improve what can be achieved locally. Collaborative Clustering also respects the security constraint because no native information (meaning no information contained in a specific view) is exchanged during inter-view communications. The communication is performed by exchanging either individuals affectations to their clusters [13], or indirect information used by each clustering algorithms during its learning phase. Several works are also focused on how to choose the best views to collaborate with [4], which can be compared to the Masked Weighting Method introduced in this paper.

However, Collaborative Clustering and the Collaborative Reconstruction System differ regarding their final goals. With Collaborative Clustering, the user tries to find the best consensus between several clustering algorithms or clustering partitions. With our system, the user wants to reconstruct missing data.

### C. Collaborative Filtering

The second field related to our Collaborative Reconstruction System is Collaborative Filtering. In Collaborative Filtering, a set of individuals rate a set of items with a certain sparsity, meaning that each individual rates only a small percentage of the items available. The aim of Collaborative Filtering is therefore to predict what would be the rating of an individual for some items knowing all the other individuals ratings. In [14] as well as in [15], the authors present the state of the art and the challenges of Collaborative Filtering. The system presented in this article can be considered as a multi-view alternative to Collaborative Filtering. Moreover, the fact that Collaborative Filtering methods use raw available data to make their predictions does not respect the security constraint. One can also notice that in the past few years, some works have focused on the usability of Neural Networks [16], [17], in particular Autoencoders [18]–[20], in Collaborative Filtering.

### D. Security

To prevent the direct identification of individual during data transfer or their use by an algorithm, two main approaches can be used: The first approach, K-Anonymity relies reducing the number of unique attribute combinations in the dataset, thus rendering the dataset records undistinguishable from each other, so that a potential attacker could not identify a unique individual.

Only recently the link between K-anonymity and dimensionality reduction has been studied. In [21], Tail et al, show that Autoencoders can provide k-anomity by dimensionality reduction. Thay also high light a trade-off between data utility and k-anonymity depending on the hidden layer size of the Autoencoder network. This work demonstrates that by using Autoencoders, a certain level of data privacy can be achieved while preserving utility of the data. Our DCRS system relies on this assumption by using Autoencoders to encode and exchange data between views.

## III. Problem definition

Let $\mathcal{X}$ be a set of individuals. Let $V_0, V_1, ..., V_n$ be a set of views, each in its own feature space $\mathcal{F}_0, \mathcal{F}_1, ..., \mathcal{F}_n$ , such that $V_i : \mathcal{X} \rightarrow \mathcal{F}_i$. Let $\mathcal{X}_i \subset \mathcal{X}$ be the subset of individuals visible in view $V_i$. In other words $\mathcal{X}_i$ is the subset of the population for which data is available in the feature set of view $V_i$. We note $V_{i|j}$ the subset of $V_i$ (in the feature space of $V_i$) which individuals are also present in $V_j$.

To its core, the *cooperative reconstruction system (CRS)* aims at learning, in view $i$, a reconstruction function $F_i$ of individuals $x \notin X_i$ in view $V_i$, based on information provided by the other views. Therefore *ie*: $F_i : \cup_{j \neq i} \mathcal{F}_j \rightarrow \mathcal{F}_i$:

$$\tilde{x}_{u,i} = F_i(x \in X_{j \neq i}) \tag{1}$$

This formulation is often used in recommender systems (see Section II), but in the context of multi-view systems, it ignores two critical constraints:

1) Data Security: in the context of this paper, data security is defined as the constraint of not being able to access original data if it is not from its original view. The input space of the reconstruction function should be different from the concatenation of the other views feature spaces.
2) Scalability: If a new view is added (rep. removed) to/from the system, how is learning the new representation affected by this change.

These two constraints provide new way to formulate the problem:

$$\tilde{x}_{u,i} = F_i(x \in E_{j\neq i}(X_{j\neq i})) \tag{2}$$

Where $E_i$ is an encoding function on $\mathcal{F}_i$. This encoding must be designed in such a way that only the view containing the individual's original features can reconstruct the values from the encoding.

## IV. Cooperative Reconstruction System

In this section, we describe the architecture of our proposed Cooperative Reconstruction System. A representation of this system can be found on Figure 1. Our system is based on several modules: first, to solve the problem of security-friendly information transfer, the system uses a set of $N$ Autoencoders [2] –with $N$ being the number of views–, to locally encode data to make them impossible to read from outside of their views. In a way, our proposal is similar to the architecture proposed in [22], but differs in the sense that we aim at multi-view reconstruction instead of a single consensus representation.

Let's consider the case where an individual has no representation in a view. Each external view can send an encoded version of its local data on this individual, resulting in the transfer of potentially $N-1$ encoded vectors. Then, any of the $N-1$ encoded external version can be used by a fully connected deep network to try to infer the features this individual would have had in the view it is missing, thus leading to potentially $N-1$ possible versions of the missing individual.

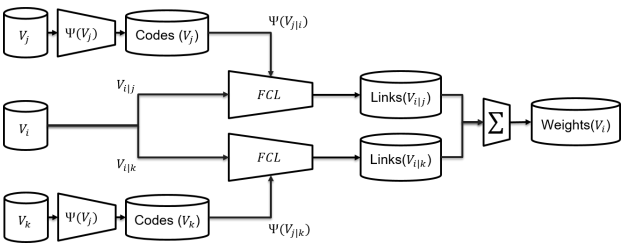



Fig. 1: Cooperative Reconstruction System. $\Psi$ represents autoencoders. In this example, Views j and k are sending their coded version of the individual to View i.

The combination of the inferred individuals can then be used to reconstruct an accurate representation of the missing individual. However, since disagreement may occur between the different $N-1$ sources of information, the inferred data from each view need to be weighted to ensure an optimal reconstruction. This is solved using a weighting method we introduce in Section IV-D and called the Masked Weighting Method. The basic idea of this method is to learn a set of $N-1$ scalar vectors, called masks, to weight each approximation generated locally (cf. Fig. 2). These masks can be trained using either Gradient Descent or using an iterative update rule. The description of both methods can be found in Section IV-D.

The global system is designed to be modular: when a new view is available, the system just has to learn its auto-encoder and the neural networks responsible for the links between this new view and the existing ones. However, due to the nature of the weighting methods between the views, all masks have to be learned again. This modularity is important because of the usually long learning time of a Deep Neural Network: learning the masks again does not take long, while having to re-train all neural networks would take a lot of time. Therefore, this modularity provides a substantial gain of time when a new view is added. This point has to be considered together with the fact that for a system made of $N$ views, approximately $N^2$ networks have to be trained.

Our system has been tested on two points: how good are the reconstructed individuals compared to their original versions, and what are the classification scores of these reconstructions compared to the original ones. Thus, we tested both its efficiency at reconstruction and whether or not reconstructed data could be used for further Machine Learning.

### A. Preconditions

We assume that for all pairs of views $i \neq j$, $V_{i|j} = \mathcal{X}_i \cap \mathcal{X}_j \neq \emptyset$. The size of this set is important because it will define the quantity of information available to train the inter-views Links (cf. Section IV-C)

### B. Autoencoders

Autoencoders are specific types of Deep Neural Networks [2] which use their data both as input and output. Their main purpose is to obtain a new representation of the input data. They can also be used as a compression method if the encoding layer length is set to be smaller than the number of features describing the original data [2]. Formally, an Autoencoder is trained by minimizing a loss function, in our case, the Mean Square Error (MSE). With our notations, the MSE for a view $V_i$ would be defined as follows:

$$\frac{1}{|V_i|}\sum_{x\in V_i}(x-\hat{x})^2 \tag{3}$$

With $\hat{x}$ being the output of the Autoencoder used in the $i$-th view for the input vector $x$ and $|V_i|$ being the number of elements of $V_i$.

We have selected Autoencoders to transfer information from a view to another because they offer two advantages : First, they encode data as scalar values, which allows to use the codes as input for further analysis, and second, they make it

difficult to retrieve the original data without their decoding part, thus limiting possibilities of security breach. These properties made autoencoders widely used in the literature for other similar deep multi-view learning methods. Moreover, in our setting the Autoencoders used in each view do not need to have the same architecture nor code lengths. This flexibility allows each view to use the best encoding architecture to describe their data.

When all the Autoencoders are trained, each view $j$ is able to encode the subset $V_{j|i}$ of its dataset $V_j$, before sending the result to every other view $i$ it has to collaborate with.

### C. Links

A Link is a Neural Network in charge of infering the values of missing individuals based on the encoded data it received from an external view. In this article, a Link is more specifically a fully connected multi-layer network: to reconstruct data in a local view $i$ given information from view $j$, the Link will be trained using the version of $V_{j|i}$ encoded by the $j$-th Autoencoder as its input, and $V_{i|j}$ the original data as its output. We remind that $V_{i|j}$ and $V_{j|i}$ are the sets of shared individuals described in $V_i$ and $V_j$ feature spaces respectively, so they necessarily represent the exact same set of individuals.

It has to be noted that the receiving view $j$ never tries to decode the encoded version of $V_{i|j}$, it only tries to infer the individuals features used in its local view. This latter point is important because it is the one that ensure the security provided by the system.

In some cases, it may happen that $V_{i|j} = \{\emptyset\}$, or is not big enough to learn the link between views $i$ and $j$. The modularity of the method presented here implies that in this case, the information coming from the external view $j$ is not taken into account, and the local view $i$ will reconstruct its missing individuals based on the information from the other external views.

As this case does not change the global method, for the rest of the paper we will only consider the case in which the individuals used in the training sets are present in all views. This simplification only aims at clarifying future algebra presented in Section IV-D. When all the Links have been trained, each view has access to (at most) $N-1$ Links allowing it to infer (at most) $N-1$ version of the missing individual values.

### D. Masked Weighting Method

When a local view $i$ is missing a data and when it has access to the $N-1$ inferred versions of this missing data, $\{x_{i|j}, j \in [1..N] \setminus i\}$, it is necessary to find an efficient way to combine them to get the final result. We present a method based on a set of scalar vectors $W_i = \{w_{i|j}, j \in [1..N] \setminus i\}$ such that $w_{i|j}$ is of same dimension as vectors of $V_i$. To get the final output $\widetilde{x}_i$ of the system in the local view $i$, we use the following formula:

$$\widetilde{x}_i = \sum_{j \in [1..N] \setminus i} x_{i|j} \otimes w_{i|j} \tag{4}$$

with $\otimes$ the pointwise vector product and $x_{i|j}$ the version of the missing data inferred using data from the view $j$.

The coefficients are first initialized using equal weights summing to 1 for all features. Then, they can be learned using two methods : either using gradient descent on the reconstruction error, or through an iterative update using the zero of the derivate of this latter error. The analytical description and the characteristics of each method are described in the following section.

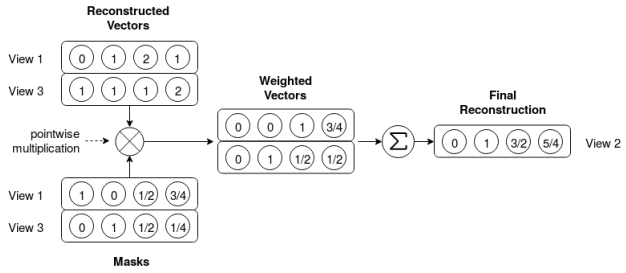



Fig. 2: The Masked Weighting Method. View 2 has got the reconstructed individuals from Views 1 and 3, and it uses the masks previously trained to get the final weighted result.

*1) Gradient Descent:* Using the system output, it becomes possible to perform a Gradient Descent on the parameters of $W_i$. In this paper, the error being used is the MSE between target data and reconstructed ones. The computation of the error $E_i$ for the view $i$ can be written as follows:

$$E_i = \frac{1}{|V_i|} \sum_{x_i \in V_i} \sum_{k=1}^{\dim(V_i)} (x_i^k - \sum_{j \in [1..N] \setminus i} w_{i|j}^k x_{i|j}^k)^2 \tag{5}$$

where $x_i^k$ is the $k$-th coordinate of the individual $x_i$. The differentiation of $E_i$ w.r.t. the parameters $w_{i|j}^k$ of $W_i$ can then be written:

$$\frac{\partial E}{\partial w_{i|j}^k} = \frac{2}{|V_i|} \sum_{x_i \in V_i} x_{i|j}^k \left( \widetilde{x}_i^k - x_i^k \right) \tag{6}$$

This latter formula makes it possible to update the weight $w_{i|j}^k$ using the usual gradient formula

$$(w_{i|j}^k)^{new} = (w_{i|j}^k)^{old} - \epsilon \frac{\partial E}{\partial w_{i|j}^k} \tag{7}$$

where $\epsilon > 0$ is the parameter defining the learning rate of the process. This update process is performed on every weight until convergence. In practice, the learning is stopped when the norm of the update value defined in Eq. 6 goes under a threshold fixed by the user.

*2) Iterative update:* : It is also possible to update weights based on the minimum of $E_i$ found using Eq.6, which after a few developments gives us:

$$\frac{\partial E_i}{\partial w^k_{i|j}} = 0$$
$$\Rightarrow \quad w^k_{i|j} = \frac{\sum_{x_i \in V_i} x^k_{i|j} \left(x^k_i - \sum_{j' \in [1..N]\setminus\{i,j\}} w^k_{i|j'} x^k_{i|j'}\right)}{\sum_{x_i \in V_i} \left(x^k_{i|j}\right)^2} \tag{8}$$

Eq.8 shows that the update of $w^k_{i|j}$ requires the values of $\{w^k_{i|j'}, j' \in [1..N]\setminus\{i,j\}\}$. Thus it is possible to define an iterative update for which the values of $\{w^{k,t}_{i|j'}, j' \in [1..N]\setminus\{i,j\}\}$ at time $t$ are used to obtained $w^{k,t+1}_{i|j}$ at time $t+1$. This problem being convex, the iterative process is performed until convergence of the weights.

This weighting method is used because it offers several advantages:

1) With either a noisy external view or a low-quality Link, the weighting coefficients for this view will converge to a value under $\frac{1}{N-1}$ which is the value corresponding to a mean of the external views. By doing so, the method lowers the impact of the bad reconstruction on the result.
2) On the opposite, this method will favor views which might greatly improve the final reconstruction with a weight over $\frac{1}{N-1}$.
3) Contrary to a weighted mean which would assign a single scalar to a view, this method allows to favor only a subpart of an inferred vector. One can easily imagine that an external view would only allow to recover parts of the local information (see the Cube dataset V-A). Our weighting method makes it possible to automatically identify these parts during parameters training.

When $W_i$ has been trained for all the views, the system is ready to use on missing data. An abstraction of the reconstruction process can be found on Figure 3, and a summary of the system architecture can be found on Figure 1. This latter can be used to attest that there is no original data going from a local view to an external one.

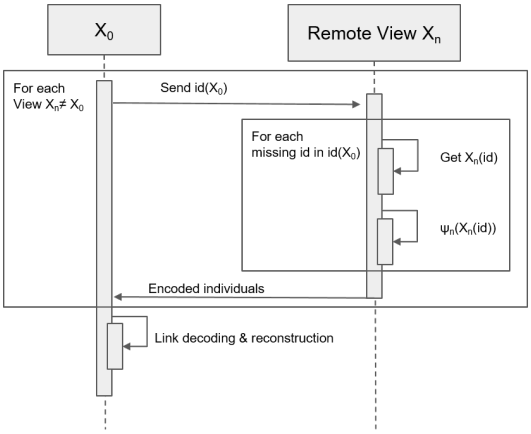



Fig. 3: Reconstruction process: Identification of a missing item, encoding in the remote view, and reconstruction in the local view.

## V. Experimental setting

This Section presents the experiments that have been conducted to test our proposed method. In Section V-A we detail the datasets used for our experiments, then the global methodology used to analyze the system behavior is described in Section V-B. The measures used to quantify the results are presented in Section V-C, and finally numeric results are presented in Section VI.

### A. Datasets

We used the following 3 datasets:

1) *Wisconsin Diagnostic Breast Cancer (WDBC)*: This dataset has 569 instances with 32 variables (ID, diagnosis, 30 real-valued input variables). Each data observation is labeled as benign (357) or malignant (212). Variables are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of 3 cell nuclei, so we have 3 natural views.
2) *Multi-Features Digital Dataset (MFDD) [1]*: This dataset consists of features of handwritten numerals (from 0 to 9) extracted from a collection of Dutch utility maps. 200 patterns per class (for a total of 2,000 patterns) have been digitized in binary images. These digits are represented in terms of the following six feature sets, each set being here used as a view: 76 Fourier coefficients of the character shapes, 216 profile correlations, 64 Karhunen-Love coefficients, 240 pixel averages in $2 \times 3$ windows and 47 Zernike moments morphological features. Each set of coefficient stands for a view.
3) *Madelon*: This dataset is an artificial dataset containing 4400 data points with 500 features. The data can be grouped in 32 clusters placed on the vertices of a five dimensional hypercube and randomly labeled +1 or -1. The five dimensions constitute 5 informative features. 15 linear combinations of those features were added to form a set of 20 (redundant) informative features. Based on those 20 features one must separate the examples into the two classes (corresponding to the +-1 labels). Finally 480 features called 'probes' having no predictive power were added by the authors. The order of the features and patterns is random. This dataset is the most challenging among these used in this article. It is used to test the ability of our sytem to ignore noise (it should not reconstruct it) and to show that despite the large number of noisy features, we still have good classification results regardless of the poor reconstruction. Because no further information in available on this dataset, the views are randomly generated by picking a random set of 125 features for each.
4) *Cube*: In addition of the three previously described datasets, we have created a toy example which we mainly use to test the effectiveness of the Masked Weighting Method. This dataset, which we will refer to as the Cube

dataset, is made of 1000 3-dimensional points divided in 4 classes of 250 members each. The points of each class are generated using a normal law with a standard deviation of $0.1$ and centered either on the center of the feature space $(0, 0, 0)$, or at the extremity of one on the three unit vectors $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. A graphical representation of the Cube dataset can be seen on Figure 4. The 3 views are obtained by projecting the whole dataset according to one of the three previous unit vectors. The point of this segmentation is explained in Section VI.

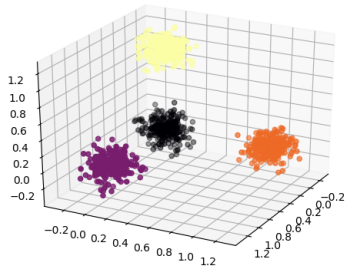


Fig. 4: The Cube dataset

### B. Methodology

In order to analyze each aspect of the system, several sets of experiments have been conducted. The first set consists in training a system with and without the Masked Weighting Method and to analyze the results in terms of reconstruction quality (Section VI-A). When it comes to combining the results from each external views, the system without our combination method simply uses a normalized equi-weighted sum of the reconstructed external vectors. This first experiment has been conducted in order to both test the viability of the method and determine the impact of our new combination method. An intermediary result is presented in Section VI-B: the reconstruction of images from the MFDD dataset is graphically presented. We use this reconstruction as an example of our multi-view reconstruction system: based on its training and informations from the other views (the 76 Fourier coefficient, profile correlations, Karhunen-Love coefficient, the Zernik moments, or any combinations of theses), our algorithm attempt to reconstruct from scratch the accurate pixel representation of the digits.

The second set of experiments consists of the analysis of the results obtained during the first set, but this time considering the impact of the Masked Weighting Method on a classification task performed on the reconstructed data (Section VI-C). Finally, the third and last set of experiments consists of the analysis of the masks values for the toy example (Section VI-D). This is done to ensure the method is able to determine which reconstructor is better for which part of the reconstructed individuals. Gradient descent optimization is used in the last experiments.

For all sets of experiments, the global methodology remains the same: each view is split in a training set (90%) and a test set (10%), then all neural networks (Autoencoders and Links) are trained using the required training set. To test the system, the process described in Figure 3 has been conducted on the test dataset of each view, with the results being compared to the original data.

As there might be some variability in the results depending on the initialization of each neural network, the experiments have been conducted several times and the results have been averaged. Experiments on the WDBC dataset were repeated 50 times, while these performed on MFDD 20 times, these on Madelon 10 times and these on Cube 50 times. This difference is due to different dataset sizes, which increases the training time necessary for each neural network.

In Table I above, we specify the architecture that we used to each dataset, with $NF$ the number of features in the views. The numbers in this table are the number of units per layer for the autoencoders and the links. All activation functions are "ReLu".

### C. Measures

To determine the performance of our system, we used three measures. The first one is the Mean Squared Error (MSE) between the reconstructed vector and its target. Given two $K$-dimensional vectors $x$ and $y$ with respective coordinates sets $\{x_i\}_{i\in[1..K]}$ and $\{y_i\}_{i\in[1..K]}$, their MSE can be computed as follow:

$$MSE(x, y) = \frac{1}{K}\sum_{i=1}^{K}(x_i - y_i)^2 \tag{9}$$

The global error is then the average of the MSE of all the reconstructed vectors compared with their target values. The point of this measure is to get a global idea of the distance between the reconstructed vectors and the target ones.

The second error we use is the Mean Relative Difference (MRD) between the feature values of the reconstructed vector and these of the target vector. Given the same $x$ and $y$ than above, their MRD is computed as follow:

$$MRD(X, Y) = \frac{1}{K}\sum_{i=1}^{K}\left|\frac{x_i - y_i}{y_i}\right| \tag{10}$$

Here again, the global error is the average of the MRD of all the reconstructed vectors compared to their target values. This measure is used pairwise with the MSE in order to get more precise information about the difference between the reconstructed vector and the target one. Because of the security constraint and because of the difficulties the system may have to link the views, we do not expect these errors to be as good as these obtained by reconstruction and inference systems with less constraints such as standard Multi-Layer Peceptron [23].

We test the usability of the reconstructed vectors on a classification task: Random Forest classifiers were trained on the original data (one for each view), then we tested whether or not the data reconstructed using our proposed method were classified correctly. The results were compared with

TABLE I: Neural network configuration

| Dataset | Cube | | | Madelon | | | MFDD | | | WDBC | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Autoencoders | $NF$ | 5 | $NF$ | $NF$ | 200 | $NF$ | $NF$ | 150 | $NF$ | $NF$ | 15 | $NF$ | |
| Links | 5 | 20 | $NF$ | 200 | 100 | $NF$ | 150 | 150 | $NF$ | 15 | 15 | 10 | $NF$ |

performances on a test set with complete non reconstructed data.

The error considered here is the mean difference between the classification scores obtained in each view on their test datasets with the original data and the ones obtained with the reconstructed individuals. For the remainder of the paper, we will name this error *the Classification Difference*. Contrary to the two previous ones, this error is not intended to determine the difference between a vector and its reconstruction, but rather to look at the impact of the reconstruction process on later data processing (such as a classification task). Even with mitigated reconstruction scores (MSE and MRD), a low Classification Difference would mean that the reconstructed individuals can be used in further applications. This score is presented along with the classification scores of each view. The Random Forest classifiers were trained using the entropy cost function, with 50 estimators and with a max depth of 5.

Finally, to ensure the efficiency of the Masked Weighting Method, we simply analyzed the vectors values of these masks for the Cube dataset. This dataset is particular because the projection performed to obtain a view entails the overlap of 2 clusters around the point $(0, 0)$. Moreover, projecting according to a specific axis, which is equivalent to supress a column in the original 3-dimensional dataset, prevents the local view to have any information on this axis, while its pairs will need this information to reconstruct their local individuals. If the Masked Weighting Method works as intended, a huge difference between the values of the mask should be observed. This process is illustrated in Figure 5.

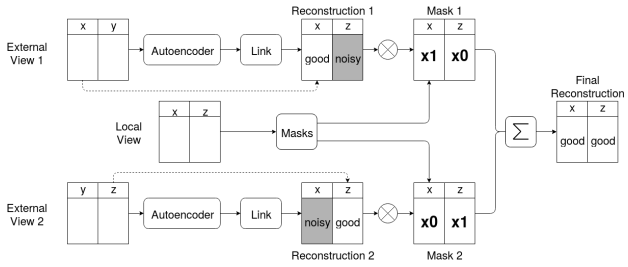



Fig. 5: The combination of two partially good reconstructions into a good one. In this example, each view has enough information to reconstruct only one feature out of the two in the local view (doted lines). The Masked Weighted Method is designed to favor the best reconstructed part of each partial reconstruction, hence the $\times 0$ and $\times 1$ in the masks.

## VI. Results

This Section is divided following the different kind of experiments that were conducted. Section VI-A presents the numeric results of the reconstruction process. Section VI-B details some visual results showing the quality of the reconstructed individuals using the MFDD database. Section VI-C presents the results obtained on the classification process performed on the reconstructed individuals, and finally Section VI-D presents the analysis conducted on the evolution of the masks coefficients depending on the information shared by views.

### A. Basic reconstruction with and without the Masked Weighting Method

During this experiment, we were interested in the impact of our combination method on the results of the system. A summary of the results on WDBC, MFDD, Madelon and Cube can be found in Figure 6.

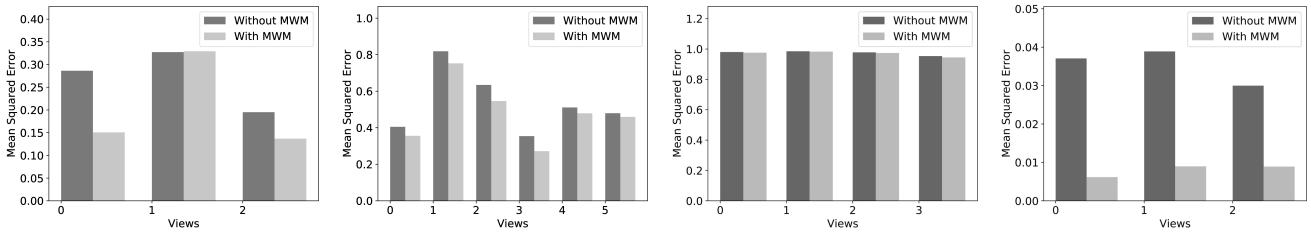
For WDBC, MFDD and Cube, the Masked Weighting Method significantly reduces the MSE for almost every view (Figures 6a, 6b and 6d). This was expected because the use of this method implies the optimization of parameters w.r.t. this error. Moreover, the MRD is reduced for all the views in WDBC, MFDD and Cube (Figures 6e, 6f and 6h): the reconstructed individuals are closest to their original versions. The exceptionnal results obtained for the MSE on the Cube dataset (Figure 6d) can be explained by the fact that this dataset has been created as a perfect example for our weighting method. Further results can be found in Section VI-D.

Considering the reconstruction results on the Madelon dataset (Figure 6c and Figure 6g), the high MSE and MRD values were expected because of the numerous noisy features present in every view (480 out of 500): the Links could not reconstruct noise based on some more noise. The values around 1 for the MSE and MRD (Figure 6c and 6g) can be explained by the fact that during the training, the trained Links were only returning values around $10^{-2}$ (the noise could not be reconstructed as expected), while the scaled dataset mostly consists of values around 1. This case presents an extreme situation for which our system does not work as intended: the fact that it tries to reconstruct every feature of the local view implicitly implies that these features are not too noisy and can also be explained using the information available in the external views, which is not the case for the Madelon dataset.

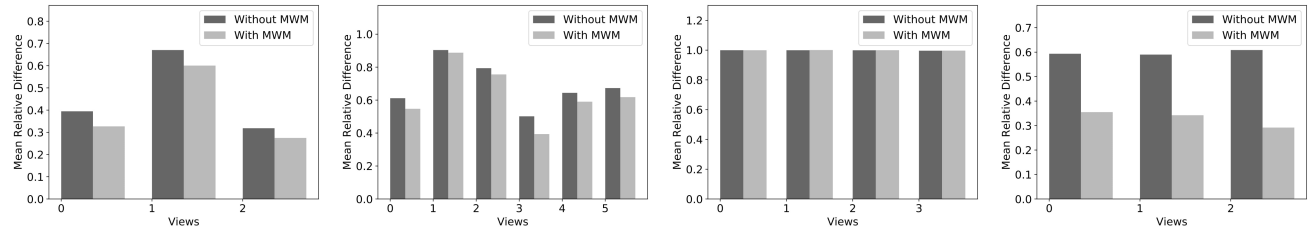
### B. Graphical Reconstruction of Handwritten Digits

To better analyze the quality of the reconstructed individuals, we have used a specific view of the MFDD dataset, namely the one with the 240 pixel averages in $2 \times 3$ windows. The individuals of the tests datasets have been reconstructed and plotted.

While the MSE and the MRD are high for this reconstruction (Figure 6b and Figure 6f), the reconstructed individuals can be easily recognized as shown in Fig. 7.

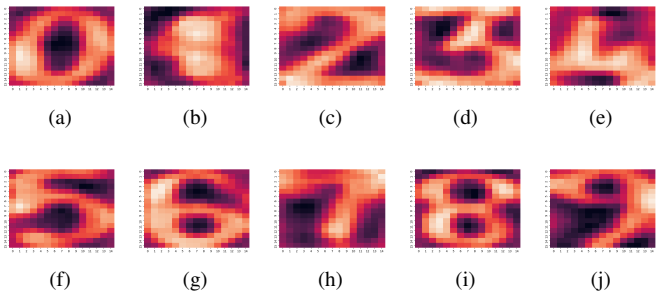



(a) WDBC (b) MFDD (c) Madelon (d) Cube




(e) WDBC (f) MFDD (g) Madelon (h) Cube

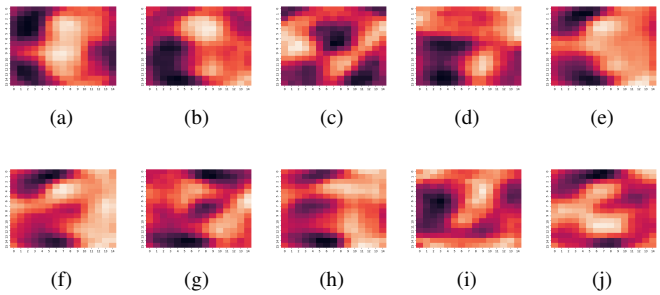
Fig. 6: Mean Squared Error and Mean Relative Difference for all the datasets. A lower value corresponds to a better result.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 7: Sample of the reconstructed images available in the MFDD dataset. Some well reconstructed examples.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 8: Sample of the reconstructed images available in the MFDD dataset. Some poorly reconstructed examples.

However, while it is true for most of the reconstructed images, some examples do not work as well, as presented in Figure 8. Moreover, even if one can recognize the numbers, a blurring effect is observed even on the best reconstructed examples. This highlights that the system tends to remove noise from the data it reconstruct, which is a good thing, but also that it may sometimes be suffering from overfitting and reconstructing data based mostly on the most discriminant elements, thus resulting in a good classification of poorly reconstructed data. We can imagine, that especially in the case of image reconstruction this can be problematic.

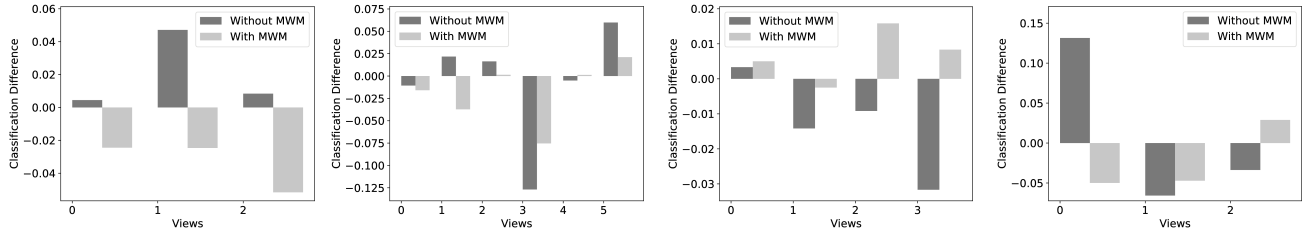
### *C. Impact on the Classification Accuracy*

For the second experiment, one needs to refer at Table II and Figure VI-C. Figure VI-C is obtained by substracting the accuracies obtained on the original data to the accuracies of the system with and without the MWM. We notice that the Collaborative Reconstruction System gives a classification accuracy comparable to these obtained on the original data: for WDBC, MFDD and Cube, the maximum absolute value of the Classification Difference is 7.5% (for the version using the Masked Weighting Method) when the mean original scores are respectively 90.9%, 88.4% and 73,35%. Secondly, Figure VI-C shows that for a majority of views, our combination method tends to lower the absolute Classification Difference.

TABLE II: Mean classification rate per database on the original data.

| **Dataset** | WDBC | MFDD | Madelon | Cube |
|---|---|---|---|---|
| **Mean rate** | 0.909 | 0.884 | 0.606 | 0.733 |

Even if we do not have clearly identified the source of this phenomenon, we suggest the following explanation: the quality of the output of a reconstruction system which does not use our combination method is highly dependent on the quality of the Links which make the inter-view reconstruction possible. Even if many tests have been performed for each




(a) WDBC (b) MFDD (c) Madelon (d) Cube

Fig. 9: Classification Accuracies relative to the classification rates obtained on whole datasets for WDBC, MFDD, Madelon and Cube: classification scores from the reconstructed data without and with the Masked Weighting Method and their differences.

database, the results depends on both manageable (hyperparameters of all the neural networks) and unmanageable (local minimum, initialization) points, both being very sensitive for the system training. That being said, it is very likely that the system results are very sensitive, which would explain the higher variability of the results obtained without the combination method compared to these obtained with it. This latter probably tends to mitigate the variability of the results because it depends far less on sensitive points: it only requires a learning step if the gradient descent method is used to update the weights and the initialization is the same every time. This phenomenon is particularly visible for the Cube dataset: while every view represents approximately the same thing, results without the MWM are more variable than those with.

### *D. Adaptation of the masks coefficients*

The point of this last set of experiments was to analyze the evolution of the masks coefficients to ensure that the method was able to determine which part of each reconstructed vector was the most useful to reconstruct the final individual. To make that possible, the Cube dataset was generated as explained in Section V-A, leading to the creation of 3 views each defined by 2 features. For each view, one of its feature is shared by one of the external views and the other feature is shared by the other external view. The point of this structure is to limit the mutual information that two views can share. If the mutual information is limited to a specific set of features (the set being composed of only one feature in this example), the quality of the partial reconstructions should vary depending on the reconstructed feature, as presented in Figure 5.

In the Cube example, the information is either totally shared (same values if the feature is present in both views) or not at all (the feature not being present in the external view). Thus, we expect to obtain mask values around respectively 1 and 0. The results obtained empirically are described in Table III. It clearly appears that the masks values adapt depending on the feature they are weighting: while these linked to the shared features are above 0.9, the ones linked to the other features never exceed 0.15. This validates the efficiency of the masks adaptation depending on the mutual information.

TABLE III: Mean and standard deviation of the values of the masks coefficients depending on the feature they are weighting

| | Mean | Standard deviation |
|---|---|---|
| **Shared feature** | 0.920 | 0.026 |
| **Non shared feature** | 0.143 | 0.034 |

## VII. Conclusion & Perspectives

In this paper, in a global context of multiplication of multi-view data, we have presented a new system called the Cooperative Reconstruction System. The purpose of this system is to reconstruct data missing in some views by using information contained in other views. We do so without sharing the original data, thus avoiding security issues. To do this, the system relies on three modules: Autoencoders to encrypt the data under a compressed scalar vector form, fully connected deep networks -called Links- to decipher an external code in a local view, and the Masked Weighting Method, a new weighting method to combine all external reconstructions, thus obtaining the final reconstruction.

The Masked Weighting Method has 3 functions: combining external information, reducing the influence of views with information which could hinder the reconstruction process, and reducing the impact of missing data during the system training process.

The efficiency of both our reconstruction system and our combination method has been tested on four different datasets: WDBC, MFDD, Madelon and Cube. To this end, two criterion have been considered: the adequation of the reconstructed individuals to their original versions considering using the Mean Squared Error and the Mean Relative Difference, and the impact of the use of reconstructed individuals instead of the original ones for classification purposes (tested against Random Forests in this paper). These experiments have demonstrated the main strengths and weaknesses of the system. Its main strengths are its ability to reconstruct an individual usable in a classification task without sharing data between views as well as its ability to weight views in such a way that it improves the final result compared to a standard meaning of the external reconstructions. On the opposite, its main weaknesses are its relatively weak reconstruction scores. This

is due to our system both removing noise and reconstructing mostly based on the most discriminative characteristics of each classes, thus leading to a good classification accuracy, even with data that are poorly reconstructed.

As future works, we plan on improving the reconstructions acquired from the external views through the modification of the inter-view Links. Likewise, because of the potentially high dimensionality, the use of another error than the MSE should be considered. A feature selection process may be added to the system, thus limiting the impact of the noise features in the original dataset. Another possible future extension of this work would be to work on a lighter architecture that would scale better with large datasets, or to work on an online version to alleviate the issue of scaling to large datasets.

## REFERENCES


[1] M. P. W. van Breukelen, D. M. J. Tax, and J. E. den Hartog, "Handwritten digit recognition by combined classifiers,," *Kybernetika*, vol. vol. 34, pp. 381–386, 1998.

[2] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[3] J. Sublime, D. Maurel, N. Grozavu, B. Matei, and Y. Bennani, "Optimizing exchange confidence during collaborative clustering," in *The 2018 International Joint Conference on*. IEEE, 2018.

[4] J. Sublime, G. Cabanes, and B. Matei, "Study on the influence of diversity and quality in entropy based collaborative clustering," *Entropy*, vol. 21, no. 10, p. 951, 2019.

[5] M. C. P. de Souto, P. A. Jaskowiak, and I. G. Costa, "Impact of missing data imputation methods on gene expression clustering and classification," *BMC Bioinformatics*, vol. 16, pp. 64:1–64:9, 2015.

[6] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "On deep multi-view representation learning: Objectives and optimization," *CoRR*, vol. abs/1602.01024, 2016.

[7] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, L. Getoor and T. Scheffer, Eds. Omnipress, 2011, pp. 689–696.

[8] G. Andrew, R. Arora, J. A. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, ser. JMLR Workshop and Conference Proceedings, vol. 28. JMLR.org, 2013, pp. 1247–1255.

[9] A. P. S. Chandar, S. Lauly, H. Larochelle, M. M. Khapra, B. Ravindran, V. C. Raykar, and A. Saha, "An autoencoder approach to learning bilingual word representations," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 1853–1861.

[10] P. L. Lai and C. Fyfe, "A neural implementation of canonical correlation analysis," *Neural Networks*, vol. 12, no. 10, pp. 1391–1397, 1999.

[11] ——, "Kernel and nonlinear canonical correlation analysis," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, IJCNN 2000, Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, July 24-27, 2000, Volume 4*. IEEE Computer Society, 2000, p. 614.

[12] D. R. Hardoon, S. Szedmák, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[13] J. Sublime, B. Matei, G. Cabanes, N. Grozavu, Y. Bennani, and A. Cornuéjols, "Entropy based probabilistic collaborative clustering," *Pattern Recognition*, vol. 72, pp. 144–157, 2017.

[14] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender systems handbook*. Springer, 2015, pp. 77–118.

[15] Y. Shi, M. Larson, and A. Hanjalic, "Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 3, 2014.

[16] Y. Zheng, B. Tang, W. Ding, and H. Zhou, "A neural autoregressive approach to collaborative filtering," vol. 48, pp. 764–773, 20–22 Jun 2016.

[17] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[18] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 111–112.

[19] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 2016, pp. 153–162.

[20] F. Strub and J. Mary, "Collaborative filtering with stacked denoising autoencoders and sparse inputs," in *NIPS workshop on machine learning for eCommerce*, 2015.

[21] B. Tai, S. Li, Y. Huang, N. Suri, and P. Wang, "Exploring the relationship between dimensionality reduction and private data release," in *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)*, 2018, pp. 25–33.

[22] Z. Fang, S. Zhou, and J. Li, "Multi-view autoencoder for image feature learning with structured nonnegative low rank," in *2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7-10, 2018*. IEEE, 2018, pp. 4033–4037.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.