



**HAL**  
open science

## Generalised Mutual Information: a Framework for Discriminative Clustering

Louis Ohl, Pierre-Alexandre Mattei, Charles Bouveyron, Warith Harchaoui, Mickaël Leclercq, Arnaud Droit, Frédéric Precioso

► **To cite this version:**

Louis Ohl, Pierre-Alexandre Mattei, Charles Bouveyron, Warith Harchaoui, Mickaël Leclercq, et al.. Generalised Mutual Information: a Framework for Discriminative Clustering. 2023. hal-04198445

**HAL Id: hal-04198445**

**<https://hal.science/hal-04198445>**

Preprint submitted on 7 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Generalised Mutual Information: a Framework for Discriminative Clustering

Louis Ohl, Pierre-Alexandre Mattei, Charles Bouveyron, Warith Harchaoui, Mickaël Leclercq, Arnaud Droit, Frédéric Precioso

## Abstract

In the last decade, recent successes in deep clustering majorly involved the Mutual Information (MI) as an unsupervised objective for training neural networks with increasing regularisations. While the quality of the regularisations have been largely discussed for improvements, little attention has been dedicated to the relevance of MI as a clustering objective. In this paper, we first highlight how the maximisation of MI does not lead to satisfying clusters. We identified the Kullback-Leibler divergence as the main reason of this behaviour. Hence, we generalise the mutual information by changing its core distance, introducing the Generalised Mutual Information (GEMINI): a set of metrics for unsupervised neural network training. Unlike MI, some GEMINIs do not require regularisations when training as they are geometry-aware thanks to distances or kernels in the data space. Finally, we highlight that GEMINIs can automatically select a relevant number of clusters, a property that has been little studied in deep discriminative clustering context where the number of clusters is a priori unknown.

## Index Terms

Clustering, discriminative clustering, unsupervised learning, information theory, mutual information, machine learning



- 
- L. Ohl, P-A. Mattei, C. Bouveyron and F. Precioso are with the Université Côte d'Azur, INRIA Maasai team, CNRS
  - M. Leclercq, A. Droit and L. Ohl are with the Université Laval, CHU de Québec Research Centre
  - W. Harchaoui is with Jellysmack, Artificial Intelligence Lab in Paris

*Manuscript under review*

# Generalised Mutual Information: a Framework for Discriminative Clustering

## 1 INTRODUCTION

CLUSTERING is a fundamental learning task which consists in separating data samples into several categories, each named cluster. This task hinges on two main questions concerning the assessment of correct clustering and the actual number of clusters that may be contained within the data distribution. However, this problem is ill-posed since a cluster lacks formal definitions which makes it a hard problem [40].

Model-based algorithms make assumptions about the true distribution of the data as a result of some latent distribution of clusters [6]. These techniques are able to find the most likely cluster assignment to data points. These models are usually generative, exhibiting an explicit assumption of the prior knowledge on the data.

Early deep models to perform clustering first relied on autoencoders, based on the belief that an encoding space holds satisfactory properties [23], [37], [68]. However, the drawback of these architectures is that they do not guarantee that data samples which should meaningfully be far apart remain so in the feature space. Early models that dropped decoders notably used the Mutual Information (MI) [33], [41] as an objective to maximise. The MI can be written in two ways, either as measure of dependency between two variables  $\mathbf{x}$  and  $y$ , e.g. data distribution  $p(\mathbf{x})$  and cluster assignment  $p(y)$ :

$$\mathcal{I}(\mathbf{x}; y) = D_{\text{KL}}(p(\mathbf{x}, y) || p(\mathbf{x})p(y)), \quad (1)$$

or as an expected distance between implied distributions and the overall data:

$$\mathcal{I}(\mathbf{x}; y) = \mathbb{E}_{p(y)}[D_{\text{KL}}(p(\mathbf{x}|y) || p(\mathbf{x}))], \quad (2)$$

with  $D_{\text{KL}}$  being the Kullback-Leibler (KL) divergence. Related works often relied on the notion of MI as a measure of coherence between cluster assignments and data distribution [31]. Regularisation techniques were employed to leverage the potential of MI, mostly by specifying model invariances, for example with data augmentation [37].

The maximisation of MI thus gave way to contrastive learning objectives which aim at learning stable representations of data through such invariance specifications [11], [13]. The contrastive loss maximises the similarity between the features of a sample and its augmentation, while decreasing the similarity with any other sample. Clustering methods also benefited from recent successful deep architectures [34], [43], [61] by encompassing regularisations in the architecture. These methods correspond to discriminative clustering where we seek to directly infer cluster given the data distribution. Initial methods also focused on alternate schemes, for example with curriculum learning [12] to iteratively select relevant data samples for training, by alternating K-means cluster assignment with supervised learning using

the inferred labels [10], or by proceeding to multiple distinct training steps [16], [53], [64].

However, most of the methods above rarely discuss their robustness when the number of clusters to find is different from the amount of preexisting known classes. While previous work was essentially motivated by considering MI as a dependence measure, we explore in this paper the alternative definition of the MI as the expected distance between data distribution implied by the clusters and the entire data. We extend it to incorporate cluster-wise comparisons of implied distributions, and question the choice of the KL divergence with other possible statistical distances.

Throughout the introduction of the Generalised Mutual Information (GEMINI), the contributions of this paper are:

- A demonstration of how the maxima of MI are not sufficient criteria for clustering. This extends the contribution of [62] to the discrete case.
- The introduction of a set of metrics called GEMINIs involving different distances between distributions which can incorporate prior knowledge on the geometry of the data. Some of these metrics do not require regularisations.
- A highlight of the implicit selection of clusters from GEMINIs which allows to select a relevant number of cluster during training.
- A Python package using only for clustering with GEMINI: *gemclus*

## 2 IS MI A GOOD CLUSTERING OBJECTIVE?

We consider in this section a dataset consisting in  $N$  unlabelled samples  $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ . We distinguish two major use cases of the mutual information: one where we measure the dependence between two continuous variables, as is the case in representation learning, and one where the random variable is discrete. In representation learning, the goal is to construct a continuous representation  $\mathbf{z}$  extracted from the data  $\mathbf{x}$  using a learnable distribution of parameters  $\theta$ . In clustering, samples  $\mathbf{x}$  are assigned to the discrete variable  $y$  through another learnable distribution.

### 2.1 Representation learning

Representation learning consists in finding high-level features  $\mathbf{z}_i$  extracted from the data  $\mathbf{x}_i$  in order to perform a *downstream task*, e.g. clustering or classification. MI between  $\mathbf{x}$  and  $\mathbf{z}$  is a common choice for learning features [31]. However, estimating correctly MI between two random variables in continuous domains is often intractable when  $p(\mathbf{x}|\mathbf{z})$  or  $p(\mathbf{z}|\mathbf{x})$  is unknown, thus lower bounds are preferred, e.g. variational estimators such as MINE [4],  $\mathcal{I}_{\text{NCE}}$  [63],  $\mathcal{I}_{\text{BA}}$  [3],  $\mathcal{I}_{\text{DV}}$  [18]. These bounds require more parameters: additional

discriminator networks are also trained to make a distinction between data and features issued from the joint distribution or product of marginals. Most of these lower bounds often present high-variance such as  $\mathcal{I}_{\text{NJW}}$  [50]. Poole et al. [56] eventually bridged the gap between this high-variance estimators and the high-bias low-variance  $\mathcal{I}_{\text{NCE}}$  by introducing  $\mathcal{I}_{\alpha}$ , an interpolated lower bound. Another common choice of loss function to train features are contrastive losses such as NT-XENT [13] where the similarity between the features  $\mathbf{z}_i$  from data  $\mathbf{x}_i$  is maximised with the features  $\tilde{\mathbf{z}}$  from a data-augmented  $\tilde{\mathbf{x}}_i$  against any other features  $\mathbf{z}_j$ . Recently, Do et al. [17] achieved excellent performances in single-stage methods by highlighting the link between the  $\mathcal{I}_{\text{NCE}}$  estimator [63] and contrastive learning losses. Representation learning therefore comes at the cost of a complex lower bound estimator on MI, which often requires data augmentation. Moreover, it was noticed that the MI is hardly predictive of downstream tasks [62] when the variable  $y$  is continuous, i.e. a high value of MI does not clarify whether the discovered representations are insightful with regards to the target of the downstream task.

## 2.2 Discriminative clustering

The MI has been first used as an objective for learning discriminative clustering models [8]. Associated architectures went from simple logistic regression [41] to deeper architectures [33], [37]. Beyond architecture improvement, the MI maximisation was also carried with several regularisations. These regularisations include penalty terms such as weight decay [41] or Virtual Adversarial Training (VAT) [33], [49]. Data augmentation was further used to provide invariances in clustering, as well as specific architecture designs like auxiliary clustering heads [37]. Rewriting the MI in terms of entropies:

$$\mathcal{I}(\mathbf{x}; y) = \mathcal{H}(y) - \mathcal{H}(y|\mathbf{x}) \quad (3)$$

highlights a requirement for balanced clusters, through the cluster entropy term  $\mathcal{H}(y)$ . Indeed, a uniform distribution maximises the entropy. This hints that an unregularised discrete mutual information for clustering can possibly produce uniformly distributed clusters among samples, regardless of how close they could be. We highlight this claim in section 2.3. As an example of regularisation impact: maximising the MI with  $\ell_2$  constraint can be equivalent to a soft and regularised K-Means in a feature space [36]. In clustering, the number of clusters to find is usually not known in advance. Therefore, an interesting clustering algorithm should be able to find a relevant number of clusters, i.e. perform model selection. However, model selection for parametric deep clustering models is expensive [58]. Cluster selection through MI maximisation has been little studied in related works, since experiments usually tasked models to find the (supervised) classes of datasets. Furthermore, the literature diverged towards deep learning methods focusing mainly on images, yet rarely on other type of data such as tabular data [46].

## 2.3 Maximising the MI can lead to bad decision boundaries

Maximising the MI directly can be a poor objective: a high MI value is not necessarily predictive of the quality of

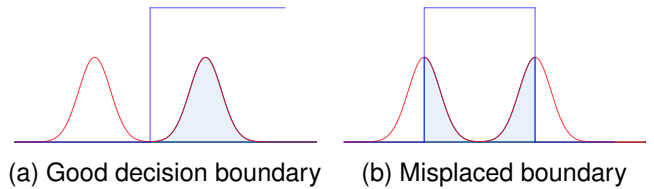


Fig. 1. Example of maximised MI for a Gaussian mixture  $\frac{1}{2}\mathcal{N}(\mu_0, \sigma^2) + \frac{1}{2}\mathcal{N}(\mu_1, \sigma^2)$ . It is clear that figure 1a presents the best decision boundary and posterior between the two Gaussian distributions. Yet, as the posterior turns sharper, the difference between both MIs converges to 0.

the features regarding downstream tasks [62] when  $y$  is continuous. We support a similar argument for the case where the data  $\mathbf{x}$  is a continuous random variable and the cluster assignment  $y$  a categorical variable. Indeed, the MI can be maximised by setting appropriately a sharp decision boundary which partitions evenly the data, i.e. when the distribution  $p_\theta(y|\mathbf{x})$  converges to a Dirac distribution. This reasoning can be seen in the entropy-based formulation of the MI (Eq. 3): any sharp decision boundary minimises the negative conditional entropy, while ensuring balanced clusters maximises the entropy of cluster proportions. Consider for example Figure 1, where a mixture of Gaussian distributions with equal variances is separated by a sharp decision boundary. We highlight that both models will have the same mutual information on condition that the misplaced decision boundary of Figure 1b splits evenly the dataset (see Appendix A).

Globally, MI misses the idea in clustering that any two points close to one another may be in the same cluster according to some chosen distance. Hence regularisations are required to ensure this constraint. An early sketch of these insights was mentioned by Bridle et al. [8] or Corduneanu et al. [14]. The non-predictiveness of MI was as well recently empirically highlighted by Zhang and Boykov [70] in discrete cases. This can be also seen as a problem of invariance of the conditional distribution in low density areas [14].

## 3 EXTENDING THE MI TO THE GEMINI

Given the identified limitations of MI, we now describe the discriminative clustering framework based on the expected distance equation of the mutual information. We then detail the different statistical distances we can use to extend MI to the Generalised Mutual Information (GEMINI).

### 3.1 The discriminative clustering framework for GEMINIs

We only consider two random variables: the data  $\mathbf{x}$  which can be continuous or discrete and the cluster assignment  $y$  which is discrete. Instead of viewing the mutual information as a dependence-seeking objective (Eq. 1), we view it as a clustering objective that aims at separating the data distribution given cluster assignments  $p(\mathbf{x}|y)$  from the data distribution  $p(\mathbf{x})$  according to the KL divergence:

$$\mathcal{I}(\mathbf{x}; y) = \mathbb{E}_{y \sim p(y)} [D_{\text{KL}}(p(\mathbf{x}|y) || p(\mathbf{x}))]. \quad (4)$$

To highlight the discriminative clustering design, we explicitly do not set any hypothesis on the data distribution by

writing  $p_{\text{data}}(\mathbf{x})$ . The only part of the model that we design is a conditional distribution  $p_{\theta}(y|\mathbf{x})$  that assigns a cluster  $y$  to a sample  $\mathbf{x}$  using the parameters  $\theta$  of some learnable function  $\psi$  [47]:

$$y|\mathbf{x} \sim \text{Categorical}(\psi_{\theta}(\mathbf{x})). \quad (5)$$

This learnable function  $\psi_{\theta}$  can typically be a neural network of adequate design regarding the data, e.g. a CNN, or a logistic regression. Consequently, the cluster proportions are controlled by  $\theta$  because  $p_{\theta}(y) = \mathbb{E}[p_{\theta}(y|\mathbf{x})]$  and so is the conditional distribution  $p_{\theta}(\mathbf{x}|y)$  even though intractable because we cannot compute the data distribution:

$$p_{\theta}(\mathbf{x}, y) = p_{\text{data}}(\mathbf{x})p_{\theta}(y|\mathbf{x}). \quad (6)$$

This questions how Eq. (4) can be computed. Fortunately, well-known properties of MI can invert the distributions on which the KL divergence is computed [8], [41] via Bayes' theorem:

$$\mathcal{I}(\mathbf{x}; y) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [D_{\text{KL}}(p_{\theta}(y|\mathbf{x})||p_{\theta}(y))], \quad (7)$$

which is possible to estimate. Since we highlighted earlier that the KL divergence in the MI can lead to inappropriate decision boundaries, we are interested in replacing it by other distances or divergences. However, changing it in Eq. (7) would focus on the separation of cluster assignments from the cluster proportions which may be irrelevant to the data distribution. We rather alter Eq. (4) to clearly show that we separate data distributions given clusters from the entire data distribution because it allows us to take into account the data space geometry.

## 3.2 The GEMINI

### 3.2.1 Replacing the Kullback-Leibler divergence with other distances

The goal of the GEMINI is to separate data distributions according to an arbitrary distance  $D$ , i.e. changing the KL divergence for another divergence or distance in the MI. This brings the definition of our first GEMINI, the *One-vs-All* (OvA):

$$\mathcal{I}_D^{\text{OvA}}(\mathbf{x}; y) = \mathbb{E}_{y \sim p_{\theta}(y)} [D(p_{\theta}(\mathbf{x}|y)||p_{\text{data}}(\mathbf{x}))], \quad (8)$$

as it compares the distance between the distribution of a specific cluster  $p_{\theta}(\mathbf{x}|y)$  against the entire data distribution  $p(\mathbf{x})$ . There exist other distances than the KL to measure how far two distributions  $p$  and  $q$  are one from the other. We can make a clear distinction between two types of distances, Csiszar's  $f$ -divergences [15] and Integral Probability Metrics (IPM) [60]. Unlike  $f$ -divergences, IPM-derived distances like the Wasserstein distance or the Maximum Mean Discrepancy (MMD) [24], [27] bring knowledge about the data throughout either a distance  $c$  or a kernel  $\kappa$ : these distances are geometry-aware.

### 3.2.2 $f$ -divergence GEMINIs

These divergences involve a convex function  $f: \mathbb{R}^+ \rightarrow \mathbb{R}$  such that  $f(1) = 0$ . This function is applied to evaluate the ratio between two distributions  $p$  and  $q$ , as in Eq. (9):

$$D_{f\text{-div}}(p, q) = \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} \left[ f \left( \frac{p(\mathbf{z})}{q(\mathbf{z})} \right) \right]. \quad (9)$$

We will focus on three  $f$ -divergences: the KL divergence, the Total Variation (TV) distance and the squared Hellinger distance. While the KL divergence is the usual divergence for the MI, the TV and the squared Hellinger distance present different advantages among  $f$ -divergences. First of all, both of them are bounded between 0 and 1. It is consequently easy to check when any GEMINI using those is maximised contrarily to the MI that is bounded by the minimum of the entropies of  $\mathbf{x}$  and  $y$  [26]. When used as distance between data conditional distribution  $p_{\theta}(\mathbf{x}|y)$  and data distribution  $p_{\text{data}}(\mathbf{x})$ , we can apply Bayes' theorem in order to get an estimable equation to maximise (see App. F), which only involves cluster assignment  $p_{\theta}(y|\mathbf{x})$  and marginals  $p_{\theta}(y)$  as summarised in Table 1, generalising thus the work of Bridle et al. [8]. Note that all  $f$ -divergences are maximised when the two distributions  $p$  and  $q$  have disjoint supports [45]. Common  $f$ -divergence like the KL, the squared Hellinger or the Pearson  $\chi^2$  divergence, except the total variation distance, are specific cases of the  $\alpha$ -divergence subclass. The convex function of  $\alpha$ -divergence is parameterized by a real number  $\alpha$  with:

$$f_{\alpha}(t) = \begin{cases} \frac{t^{\alpha} - \alpha t + (\alpha - 1)}{\alpha(\alpha - 1)}, & \alpha \neq 0, \alpha \neq 1, \\ t \ln t, & \alpha = 1, \\ -\ln t, & \alpha = 0. \end{cases} \quad (10)$$

However, this class of  $\alpha$ -divergence is inappropriate in some cases for clustering. Indeed, we show with Proposition 3.1 (proof in App. B) that the maximisation of  $\alpha$ -divergences can lead to any clustering of the data space with balanced clusters as the discriminative model  $p_{\theta}(y|\mathbf{x})$  converges to a Dirac distribution.

**Proposition 3.1.** *Let  $\{\mathcal{X}_k\}_{k=1}^K$  a partition of  $\mathcal{X}$  such that  $\mathbb{P}(\mathbf{x} \in \mathcal{X}_k) = \frac{1}{K}$ . Then for any  $\alpha$ -divergence with  $\alpha > 0$ , the OvA GEMINI is upper bounded by a function which only depends on the proportions of the clusters. If the clustering model follows a Dirac distribution:  $p_{\theta}(y = k|\mathbf{x}) = \mathbf{1}_{[\mathbf{x} \in \mathcal{X}_k]}$ , then the upper bound is tight and the GEMINI cannot be improved.*

It is worth mentioning in Proposition 3.1 that the proportions of the cluster  $p(y = k)$  do not matter for the specific case of  $\alpha = 2$  to achieve the global maximum, i.e. for the Pearson  $\chi^2$ -divergence. We can infer from Proposition 3.1 the specific Corollary 3.1 since the MI is a case of OvA  $\alpha$ -divergence-GEMINI with  $\alpha = 1$ . We conclude that MI maximisation is a poor objective when a discriminative model can converge to a Dirac distribution.

**Corollary 3.1.** *Let  $\{\mathcal{X}_k\}_{k=1}^K$  a partition of  $\mathcal{X}$ . Then the mutual information of a discriminative distribution  $p(y|\mathbf{x})$  is upper bounded by the entropy of  $\mathbf{x}$  and the upper bound is tight if the distribution is a Dirac model  $p(y = k|\mathbf{x}) = \mathbf{1}_{[\mathbf{x} \in \mathcal{X}_k]}$ . The highest upper bound is reached when the partition is balanced.*

### 3.2.3 IPM GEMINIs

The IPM is another class of distance that incorporates knowledge from the data through a function  $f$ :

$$D_{\text{ipm}}(p, q) = \sup_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [f(\mathbf{z})] - \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [f(\mathbf{z})], \quad (11)$$

where  $\mathcal{F}$  is a set of functions. As backpropagation through suprema could be intractable, we choose to focus on two specific variations of the IPM for the GEMINI: the MMD and the Wasserstein distance. Note however that not all Wasserstein distances are IPMs and while some of our propositions are formulated for IPMs, we consider as well the entire class of the Wasserstein distances.

The MMD corresponds to the distance between the respective expected embedding of samples from the distribution  $p$  and the distribution  $q$  in a reproducible kernel Hilbert space (RKHS)  $\mathcal{H}$ :

$$\text{MMD}(p\|q) = \|\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\varphi(\mathbf{z})] - \mathbb{E}_{\mathbf{z}\sim q(\mathbf{z})}[\varphi(\mathbf{z})]\|_{\mathcal{H}}, \quad (12)$$

where  $\varphi$  is the RKHS embedding. To compute this distance we can use the kernel trick [27] by involving the kernel function  $\kappa(\mathbf{a}, \mathbf{b}) = \langle \varphi(\mathbf{a}), \varphi(\mathbf{b}) \rangle$ . We use Bayes' theorem to uncover a version of the MMD that can be estimated through Monte Carlo using only the predictions  $p_{\theta}(y|\mathbf{x})$ .

The Wasserstein distance is an optimal transport distance. It corresponds to the minimal amount of energy to transform a distribution into another according to an energy function yielding the cost  $c$  of moving the mass of a sample from one location to another:

$$\mathcal{W}_c^d(p, q) = \left( \inf_{\gamma \in \Gamma(p, q)} \mathbb{E}_{\mathbf{x}, \mathbf{z} \sim \gamma(\mathbf{x}, \mathbf{z})} [c(\mathbf{x}, \mathbf{z})^d] \right)^{\frac{1}{d}}, \quad (13)$$

where  $\Gamma(p, q)$  is the set of all couplings between  $p$  and  $q$ ,  $c$  a distance function in  $\mathcal{X}$  and  $d$  a real positive number. Computing the Wasserstein- $d$  distance between two distributions  $p_{\theta}(\mathbf{x}|y = k)$  and  $p_{\theta}(\mathbf{x})$  is difficult in our discriminative context because we only have access to a finite set of samples  $N$ . Note that in the remainder of the paper, we will focus on the Wasserstein-1 metric. The idea of an expected Wasserstein distance was first proposed by Harchaoui [29, Eq. 48] under the one-vs-rest name with an additional cluster proportion factor. However, we found this additional factor to be not grounded enough. Moreover, we can show that for the Wasserstein-1 metric the one-vs-rest Wasserstein preliminary work [29] is equivalent to the one-vs-all Wasserstein-GEMINI. To achieve the Wasserstein-GEMINI, we instead use approximations of the distributions with weighted sums of Diracs:

$$p_{\theta}(\mathbf{x}|y = k) \approx \sum_{i=1}^N m_i^k \delta_{\mathbf{x}_i} = p_N^k, \quad (14)$$

$$\text{with } m_i^k = \frac{p_{\theta}(y = k|\mathbf{x}_i)}{\sum_{j=1}^N p_{\theta}(y = k|\mathbf{x}_j)},$$

where  $\delta_{\mathbf{x}_i}$  is a Dirac located on sample location  $\mathbf{x}_i \in \mathcal{X}$ . For the distribution  $p_{\text{data}}(\mathbf{x})$ , we set all importance weights to  $1/N$ . We state in Prop. 3.2 that this empirical estimate of the Wasserstein distance converges to the correct Wasserstein distance. These importance weights are compatible with the `emd2` function of the python optimal transport package [20] which gracefully supports automatic differentiation. We describe other maximisation strategies in Appendix G.

**Proposition 3.2.** *Let  $p(\mathbf{x}|y = k_1)$  and  $p(\mathbf{x}|y = k_2)$  two cluster distributions that we empirically approximate with importance-weighted Dirac estimators  $p_N^{k_1} = \sum_{i=1}^N m_i^{k_1} \delta_{\mathbf{x}_i}$ , resp. and*

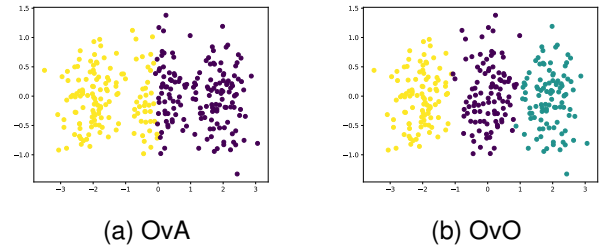


Fig. 2. Here, 3 clusters of equal proportions from isotropic Gaussian distributions are located in -2, 0 and 2 on the x-axis, with small covariance. The complete data distribution hence has its expectation in 0 on the x-axis. Consequently, maximising the OVA MMD-GEMINI with a logistic regression led to 2 clusters whereas the same model with the OvO MMD-GEMINI is able to see all 3 clusters.

$p_N^{k_2} = \sum_{i=1}^N m_i^{k_2} \delta_{\mathbf{x}_i}$ . Then the Wasserstein distance between estimates converges to the Wasserstein distance between the cluster distributions.

We refer to Appendix F.3 for proof of convergence.

### 3.3 The One-vs-One GEMINI

We question the relevance of evaluating a distance between the distribution of the data given a cluster assumption  $p_{\theta}(\mathbf{x}|y)$  and the entire data distribution  $p_{\text{data}}(\mathbf{x})$  when the geometry is taken into account. We argue that it is intuitive in clustering to compare the distribution of one cluster against the distribution of *another cluster* rather than the data distribution. Indeed, considering the geometry of the data space through a kernel in the case of the MMD or a distance in the case of the Wasserstein metric implies that we can effectively measure how two distributions are close to one another. In the formal design of the mutual information, the distribution of each cluster  $p(\mathbf{x}|y)$  is compared to the complete data distribution  $p(\mathbf{x})$ . Therefore, if one distribution of a specific cluster  $p(\mathbf{x}|y)$  were to look alike the data distribution  $p(\mathbf{x})$ , for example up to a constant in some areas of the space, then its distance to the data distribution could be 0, making it unnoticed when maximising the OVA GEMINI.

Take the example of 3 distributions  $\{p(\mathbf{x}|y = i)\}_{i=1}^3$  with respective different expectations  $\{\mu_i\}_{i=1}^3$ . We want to separate them using the OVA MMD-GEMINI with linear kernel. The mixture of the 3 distributions creates a data distribution with expectation  $\mu = \sum_{i=1}^3 p(y = i)\mu_i$ . However, if the distributions satisfy that this data expectation  $\mu$  is equal to one of the sub-expectations  $\mu_i$ , then the associated distribution  $i$  will have will not provide any information since its MMD to the data distribution is equal to 0. We illustrate this example in figure 2.

To address this issue, we introduce the second GEMINI named *one-vs-one* (OvO) in which we compare cluster distributions from independently drawn cluster assignments  $y_a$  and  $y_b$ :

$$\mathcal{I}_D^{\text{OvO}}(\mathbf{x}; y) = \mathbb{E}_{y_a, y_b \sim p_{\theta}(y)} [D(p_{\theta}(\mathbf{x}|y_a) \| p_{\theta}(\mathbf{x}|y_b))]. \quad (15)$$

The example of Figure 2 is tackled by the OvO GEMINI since the distance between each pair of the 3 clusters is non-null. Conceptually, the idea of optimising the OvO MMD-GEMINI in clustering can be found as well by França et



al. [21] who derived a regularised squared MMD in a one-vs-one setting through restrictions to Dirac distributions. Note that for most distances, the OvO GEMINI is an upper bound of the OvA GEMINI; proof of Proposition 3.3 in App. C.

**Proposition 3.3.** *Let  $D$  be an  $f$ -divergence or an IPM and  $p_\theta(y|\mathbf{x})$  a clustering distribution. Then:  $\mathcal{I}_D^{\text{OvA}}(\mathbf{x}, y) \leq \mathcal{I}_D^{\text{OvO}}(\mathbf{x}, y)$ .*

In the case of binary clustering, using an IPM distance implies equality between the OvA GEMINI and the OvO GEMINI; proof of Proposition 3.4 in App. D:

**Proposition 3.4.** *Let  $D$  be an IPM and  $p(y|\mathbf{x})$  a clustering distribution  $y$  taking  $K = 2$  values. Then:  $\mathcal{I}_D^{\text{OvA}}(\mathbf{x}, y) = \mathcal{I}_D^{\text{OvO}}(\mathbf{x}, y)$ .*

While the OvA GEMINI is maximised with Dirac clustering of the data space for some  $\alpha$ -divergence, we can extend Proposition 3.1 to all  $f$ -divergences for the OvO GEMINI with Proposition 3.5 (Proof in App. E). We notably conclude that for the total variation and the squared Hellinger distance, Dirac distributions on an even partition of the data space are the only optimal solutions.

**Proposition 3.5.** *Let  $D$  be an  $f$ -divergence,  $p_\theta(y|\mathbf{x})$  a clustering distribution such that  $p_\theta(y = k) = \frac{1}{K}$ . The OvO GEMINI is then upper bounded by a function depending only on the cluster proportions. For the upper bound to be tight, a sufficient condition is to have disjoint supports between cluster distributions  $p_\theta(\mathbf{x}|y = k)$ . The condition is necessary if the function  $f$  satisfies  $f(0) + g(0) < \infty$  where  $g(t) = tf(\frac{1}{t})$  is the convex conjugate of  $f$ .*

## 3.4 Using GEMINIs

### 3.4.1 Choosing a GEMINI

We stated in Section 3.1 that we present the GEMINI as a distance between distributions evaluated in the data space  $\mathcal{X}$  so that the distance  $D$  can take into account the topology of the data. In practice, we only design a discriminative model  $p_\theta(y|\mathbf{x})$ . Thus, we need to compute all formulas of the GEMINI through Bayes' theorem to get equations depending on  $p_\theta(y|\mathbf{x})$  and  $p_\theta(y)$ . We summarise the equations from all aforementioned GEMINIs in Table 1 (see Appendix F for derivations). We give details on the complexity of GEMINIs in Appendix H to help choose one. We propose as well some theoretical speed-ups in App. I. It is also important to consider the experimental purposes and context to choose a GEMINI. Indeed, when it is easier to design a distance than a kernel, the Wasserstein-GEMINI is more compatible than the MMD-GEMINI and vice-versa. Moreover, the MMD-GEMINI inherently computes expectations in a Hilbert Space which allows computing centroids deemed representative of the clusters. This notion of centroid is less straightforward when using the Wasserstein metric.

### 3.4.2 Estimating a GEMINI

All GEMINIs in Table 1 can be estimated using Monte Carlo making them compatible with mini-batch learning, with batch sizes of a few hundred for large datasets similarly to prior works [31], [33], [37]. We highlight the importance of the batch size when using GEMINIs. With the use of mini-batch for training, the complete GEMINI is not evaluated on the entire dataset and hence a bias may rise from the empirical estimate. This bias then has consequences on the

gradient, which in turn alters training. To illustrate this point, we generated 1000 predictions from a Dirichlet distribution with 10 clusters. These predictions are a proxy for the output of any discriminative model  $p_\theta(y|\mathbf{x})$ . We then compute the true GEMINI on all samples before evaluating it 50 times for different randomly sampled batches of increasing size. We report in Figure 3 the Mean Squared Error of all GEMINIs. We see that past 200 samples for both the OvA and the OvO models, the mean squared error is already close to or below  $10^{-2}$ , except for the OvO Wasserstein- and MMD-GEMINIs. This implies an upper bound of  $10^{-2}$  for the bias of the estimates. We conclude that there is possibly a bias in GEMINIs estimates, but it remains small enough to be negligible.

### 3.4.3 Code

The original implementation for all experiments regarding GEMINI can be found here: <https://github.com/oshillou/GEMINI>. However, later works led to the development of python package for small-scale datasets on CPU: *gemclus* at <https://gemini-clustering.github.io/>.

## 4 EXPERIMENTS

For all experiments below, we report the Adjusted Rand Index (ARI) [35], a common metric in clustering. This metric is external as it requires labels for evaluation. It ranges from 0, when labels are independent from cluster assignments, to 1, when labels are equivalent to cluster assignments up to permutations. An ARI close to 0 is equivalent to the best accuracy when voting constantly for the majority class, e.g. 10% on a balanced 10-class dataset. Regarding the MMD- and Wasserstein-GEMINIs, we used by default a linear kernel and the Euclidean distance unless specified otherwise. All discriminative models are trained using the Adam optimiser [38]. We estimate a total of 450 hours of GPU consumption. (See Appendix K for the details of python packages for experiments).

### 4.1 When the MI fails because of the modelling

We first took the most simple discriminative clustering model, where each cluster assignment according to the input datum follows a categorical distribution:

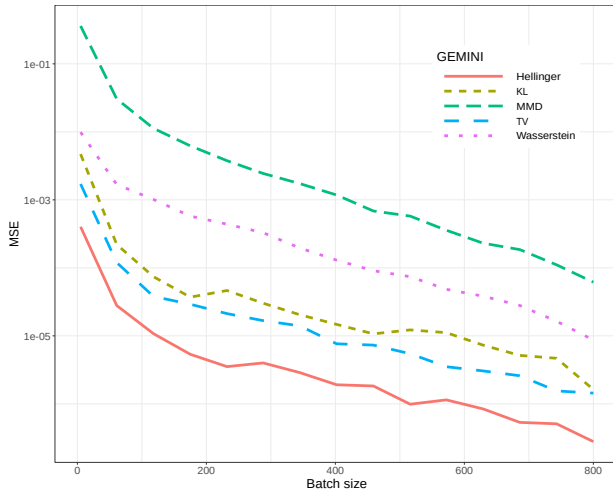
$$y|\mathbf{x} = \mathbf{x}_i \sim \text{Cat}(\theta_i^1, \theta_i^2, \dots, \theta_i^K).$$

We generated  $N = 100$  samples from a simple mixture of  $K = 3$  Gaussian distributions. Each model thus only consists in  $NK$  parameters to optimise. This is a simplistic way of describing the most flexible deep neural network. We then maximised on the one hand the OvA KL (MI) and on the other hand the OvA MMD. Both clustering results can be seen in Figure 4. We concluded that without any function, e.g. a neural network, to link the parameters of the conditional distribution with  $\mathbf{x}$ , the MI struggles to find the correct decision boundaries. Indeed, the position of  $\mathbf{x}$  in the 2D space plays no role and the decision boundary becomes only relevant with regards to cluster entropy maximisation: a uniform distribution between 3 clusters. However, it plays a major role in the kernel of the MMD-GEMINI thus solving correctly the problem.

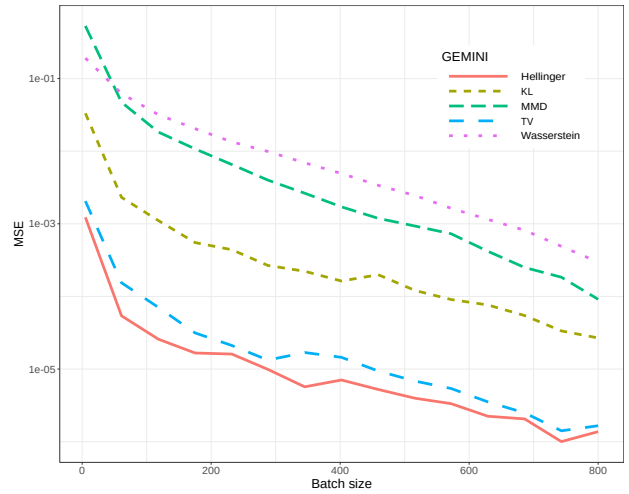
TABLE 1

Definition of the GEMINI for  $f$ -divergences, MMD and the Wasserstein distance. We directly write here the equation that can be optimised to train a discriminative model  $p_\theta(y|\mathbf{x})$  via stochastic gradient descent since they are expectations over the data.

Name	Equation
KL OvA/MI	$\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [D_{\text{KL}}(p_\theta(y \mathbf{x})\ p_\theta(y))]$
KL OvO	$\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [D_{\text{KL}}(p_\theta(y \mathbf{x})\ p_\theta(y)) + D_{\text{KL}}(p_\theta(y)\ p_\theta(y \mathbf{x}))]$
Squared Hellinger OvA	$1 - \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\mathbb{E}_{p_\theta(y)} [\sqrt{\frac{p_\theta(y \mathbf{x})}{p_\theta(y)}}]]$
Squared Hellinger OvO	$\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\mathbb{V}_{p_\theta(y)} [\sqrt{\frac{p_\theta(y \mathbf{x})}{p_\theta(y)}}]]$
TV OvA	$\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [D_{\text{TV}}(p_\theta(y \mathbf{x})\ p_\theta(y))]$
TV OvO	$\frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\mathbb{E}_{y_a, y_b \sim p_\theta(y)} [\frac{p_\theta(y_a \mathbf{x})}{p_\theta(y_a)} - \frac{p_\theta(y_b \mathbf{x})}{p_\theta(y_b)}]]]$
MMD OvA	$\mathbb{E}_{p_\theta(y)} \left[ \mathbb{E}_{\mathbf{x}_a, \mathbf{x}_b \sim p_{\text{data}}(\mathbf{x})} \left[ k(\mathbf{x}_a, \mathbf{x}_b) \left( \frac{p_\theta(y \mathbf{x}_a)p_\theta(y \mathbf{x}_b)}{p_\theta(y)^2} + 1 - 2\frac{p_\theta(y \mathbf{x}_a)}{p_\theta(y)} \right) \right]^{\frac{1}{2}} \right]$
MMD OvO	$\mathbb{E}_{y_a, y_b \sim p_\theta(y)} \left[ \mathbb{E}_{\mathbf{x}_a, \mathbf{x}_b \sim p_{\text{data}}(\mathbf{x})} \left[ k(\mathbf{x}_a, \mathbf{x}_b) \left( \frac{p_\theta(y_a \mathbf{x}_b)p_\theta(y_a \mathbf{x}_b)}{p_\theta(y_a)^2} + \frac{p_\theta(y_b \mathbf{x}_a)p_\theta(y_b \mathbf{x}_b)}{p_\theta(y_b)^2} - 2\frac{p_\theta(y_a \mathbf{x}_a)p_\theta(y_b \mathbf{x}_b)}{p_\theta(y_a)p_\theta(y_b)} \right) \right]^{\frac{1}{2}} \right]$
Wasserstein OvA	$\mathbb{E}_{p_\theta(y)} \left[ \mathcal{W}_c \left( \sum_{i=1}^N m_i^y \delta_{\mathbf{x}_i}, \sum_{i=1}^N \frac{1}{N} \delta_{\mathbf{x}_i} \right) \right]$
Wasserstein OvO	$\mathbb{E}_{y_a, y_b \sim p_\theta(y)} \left[ \mathcal{W}_c \left( \sum_{i=1}^N m_i^{y_a} \delta_{\mathbf{x}_i}, \sum_{i=1}^N m_i^{y_b} \delta_{\mathbf{x}_i} \right) \right]$



(a) OvA GEMINIs



(b) OvO GEMINIs

Fig. 3. Mean Squared Error (log scale) of estimates with varying batch sizes compared to the true value over a complete dataset of a 1000 samples. Each estimate was performed 50 times per batch size and GEMINI.

## 4.2 Resistance to outliers

To prove the strength of using neural networks for clustering trained with GEMINI, we introduced extreme samples in Gaussian mixtures by replacing a Gaussian distribution with a Student-t distribution for which the degree of freedom  $\rho$  is small. We fixed  $K = 4$  clusters, 3 being drawn from multivariate Gaussian distributions and the last one from a multivariate Student-t distribution in 2 dimensions for visualisation purposes with 1 degree of freedom or 2. Thus, the Student-t distribution produces samples that can be perceived as outliers regarding a Gaussian mixture owing to its heavy tail.

Each cluster distribution is centered around a mean  $\mu_i$  which proximity is controlled by a scalar  $\alpha$ . For simplicity, all covariance matrices are the identity scaled by a scalar  $\sigma$ .

We define:

$$\begin{aligned} \mu_1 &= [\alpha, \alpha], & \mu_2 &= [\alpha, -\alpha], \\ \mu_3 &= [-\alpha, \alpha], & \mu_4 &= [-\alpha, -\alpha]. \end{aligned}$$

To sample from a multivariate Student-t distribution, we first draw samples  $\mathbf{x}$  from a centered multivariate Gaussian distribution. We then sample another variable  $u$  from a  $\chi^2$ -distribution using the degrees of freedom  $\rho$  as parameter. Finally,  $\mathbf{x}$  is multiplied by  $\sqrt{\frac{\rho}{u}}$ , yielding samples from the Student-t distribution.

We report the ARIs of Multi-Layered Perceptron (MLP) trained 20 times with GEMINIs in Table 2. The presence of "outliers" leads K-Means and Gaussian Mixture models to fail at grasping the 4 distributions when trying to find 4 clusters. Meanwhile, GEMINIs perform better. Note that the MMD



TABLE 2

Mean ARI (std) of a MLP fitting a mixture of 3 Gaussian and 1 Student-t multivariate distributions compared with Gaussian Mixture Models and K-Means. The model can be tasked to find either 4 or 8 clusters at best and the Student-t distribution has  $\rho=1$  or 2 degrees of freedom. Bottom line presents the ARI for the maximum a posteriori of an oracle aware of all parameters of the data.

Model	$\rho = 2$		$\rho = 1$	
	4 clusters	8 clusters	4 clusters	8 clusters
K-Means	0.965 (0)	0.897 (0.040)	0 (0)	0.657 (0.008)
GMM (full covariance)	0.972 (0)	0.868 (0.042)	0 (0)	0.610 (0.117)
GMM (diagonal covariance)	<b>0.973 (0)</b>	0.862 (0.048)	0.024 (0.107)	0.660 (0.097)
$\mathcal{I}_{KL}^{ova}$	0.883 (0.182)	0.761 (0.101)	<b>0.939 (0.006)</b>	0.742 (0.092)
$\mathcal{I}_{KL}^{ovo}$	0.731 (0.140)	0.891 (0.129)	0.723 (0.114)	0.755 (0.163)
$\mathcal{I}_{H^2}^{ova}$	0.923 (0.125)	<b>0.959 (0.043)</b>	0.906 (0.103)	0.86 (0.087)
$\mathcal{I}_{H^2}^{ovo}$	0.926 (0.112)	0.951 (0.059)	0.858 (0.143)	0.887 (0.074)
$\mathcal{I}_{TV}^{ova}$	0.940 (0.097)	0.973 (0.004)	0.904 (0.104)	<b>0.925 (0.103)</b>
$\mathcal{I}_{TV}^{ovo}$	0.971 (0.005)	0.620 (0.053)	<b>0.938 (0.005)</b>	0.595 (0.055)
$\mathcal{I}_{MMD}^{ova}$	0.953 (0.060)	0.940 (0.033)	0.922 (0.004)	<b>0.908 (0.016)</b>
$\mathcal{I}_{MMD}^{ovo}$	0.968 (0.001)	0.771 (0.071)	0.921 (0.007)	0.849 (0.048)
$\mathcal{I}_{WV}^{ova}$	0.897 (0.096)	0.896 (0.021)	0.915 (0.131)	0.889 (0.051)
$\mathcal{I}_{WV}^{ovo}$	0.970 (0.002)	0.803 (0.067)	0.922 (0.006)	0.817 (0.042)
Oracle	0.991		0.989	

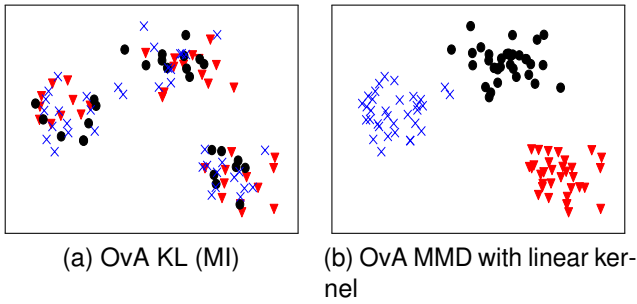


Fig. 4. Clustering of a mixture of 3 Gaussian distributions with MI (left) and a GEMINI (right) using categorical distributions. The MI does not have insights on the data shape because of the model, and clusters points uniformly between the 3 clusters (black dots, red triangles and blue crosses) whereas the MMD is aware of the data shape through its kernel.

and Wasserstein-GEMINI present lower standard deviation for high scores compared to  $f$ -divergence GEMINIs. We attribute these performances to both the MLP that tries to find separating hyperplanes in the data space and the absence of hypotheses regarding the data. Moreover, as mentioned in section 2.3, the usual MI is best maximised when its decision boundary presents little entropy  $\mathcal{H}(p_\theta(y|\mathbf{x}))$ . As neural networks can be overconfident [28], MI is likely to yield overconfident clustering by minimizing the conditional entropy. We highlight such behaviour in Figure 5 where the Rényi entropy [57] associated to each sample in the MI (Figure 5a) is much lower, if not 0, compared to OvO MMD and OvO Wasserstein (figures 5b and 5c). We conclude that Wasserstein- and MMD-GEMINIs train neural networks not to be overconfident, hence yielding more moderate distributions  $p_\theta(y|\mathbf{x})$ .

### 4.3 Leveraging a manifold geometry

We highlighted that MI can be maximised without requiring to find the suitable decision boundary. Here, we show how the provided distance to the OvO Wasserstein-GEMINI can

leverage appropriate clustering when we have a good a priori on the data.

The cluster assignment is drawn according to a Bernoulli distribution with parameter 0.5. The points are then sampled using a uniformly distributed angle in  $[0, \pi]$ . Axis symmetry is applied depending on the clusters. All points are distributed at the same radius  $\rho$  from some circle centre before adding some Gaussian noise. The samples then undergo some offset so that both moons are not linearly separable.

We generated a dataset consisting of two facing moons on which we trained a MLP using either the MI or the OvO Wasserstein-GEMINI. To construct a distance  $c$  for the Wasserstein distance, we derived a distance from the Floyd-Warshall algorithm [59], [66] on a sparse graph describing neighborhoods of samples. This distance describes how many neighbors are in between two samples, further details are provided in appendix J. We report the different decision boundaries in Figure 6. We observe that the insight on the neighborhood provided by our distance  $c$  helped the MLP to converge to the correct solution with an appropriate decision boundary unlike the MI. Note that the usual Euclidean distance in the Wasserstein metric would have converged to a solution similar to the MI. Indeed for 2 clusters, the optimal transport plan has a larger value, using a distribution similar to Figure 6a, than in Figure 6c. This toy example shows how an insightful metric provided to the Wasserstein distance in GEMINIs can lead to correct decision boundaries while only designing a discriminative distribution  $p_\theta(y|\mathbf{x})$  and a distance  $c$ .

In addition, we highlight an interesting behaviour of all GEMINIs. During optimisation, it is possible that the model converges to using fewer clusters than the number to find. For example in Figure 6, for 5 clusters, the model can converge to 4 balanced clusters and 1 empty cluster (Figure 6d) unlike MI that produced 5 misplaced clusters (Figure 6b). Indeed, the entropy on the cluster proportion in the MI forces to use the maximum number of clusters.

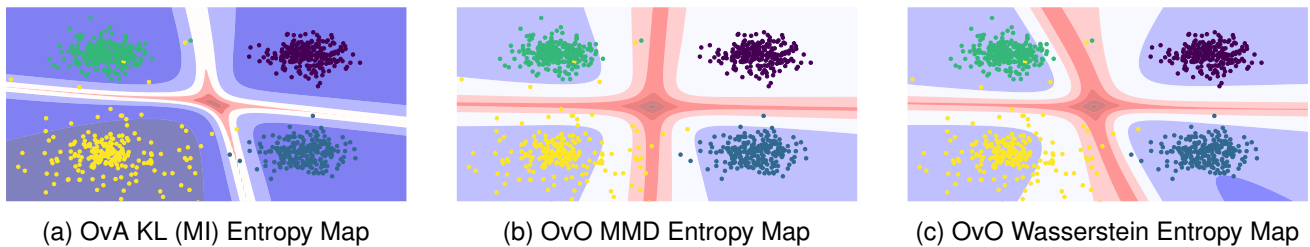


Fig. 5. Entropy maps of the predictions of each MLP trained using a GEMINI or the MI. The bottom-left distribution (yellow) is a Student-t distribution with 1 degree of freedom that produces samples far from the origin. The Rényi entropy of prediction is highlighted from lowest (blue background) to highest (red background). MI on the left has the most confident predictions overall and the smallest uncertainty around the decision boundary, i.e. high entropy variations.

TABLE 3

ARI for deep neural network trained with GEMINIs on MNIST for 500 epochs. Models were trained either with either 10 clusters to find or 15. We indicate in parentheses the number of used clusters by the model after training.

GEMINI		10 clusters		15 clusters	
		MLP	LeNet-5	MLP	LeNet-5
KL	OvA	0.320 (10)	0.138 (8)	0.271 (15)	0.136 (12)
	OvO	0.348 (7)	0.123 (4)	0.333 (8)	0.104 (4)
Squared Hellinger	OvA	0.301 (10)	0.207 (6)	0.224 (13)	0.162 (7)
	OvO	0.287 (10)	0.161 (6)	0.305 (13)	0.157 (7)
TV	OvA	0.299 (10)	0.171 (6)	0.277 (15)	0.140 (6)
	OvO	0.422 (10)	0.161 (9)	0.330 (15)	0.182 (14)
MMD	OvA	0.373 (10)	0.382 (10)	0.345 (15)	0.381 (15)
	OvO	0.361 (10)	0.373 (10)	0.364 (15)	0.379 (15)
Wasserstein	OvA	<b>0.471</b> (10)	<b>0.463</b> (10)	0.390 (15)	<b>0.446</b> (11)
	OvO	0.450 (10)	0.383 (10)	<b>0.415</b> (15)	0.414 (15)
K-Means		0.367		0.385	

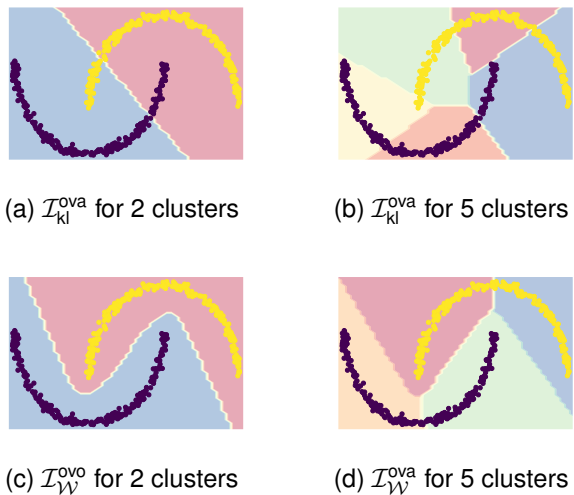


Fig. 6. Fitting hand-generated moons using the GEMINI on top of an MLP for different amount of clusters. The Wasserstein-ovo model with 5 clusters eventually found 4 clusters.

#### 4.4 Fitting MNIST

We trained a neural network using either MI or GEMINIs. Following Hu et al. [33], we first tried with a MLP with one single hidden layer of dimension 1200. To further illustrate the robustness of the method and its adaptability to other architectures, we also experimented using a LeNet-5 architecture [42] since it is adequate to the MNIST dataset.

We report our results in Table 3. Since we are dealing with a clustering method, we may not know the number of clusters a priori in a dataset. The only thing that can be said about MNIST is that there are *at least* 10 clusters, one per digit. Indeed, writings of digits could differ leading to more clusters than the number of classes. That is why we further tested the same method with 15 clusters to find in Table 3. We first see that the scores of the MMD and Wasserstein GEMINIs are greater than the MI. We also observe that no  $f$ -divergence-GEMINI always yield best ARIs. Nonetheless, we observe better performances in the case of the TV GEMINIs owing to its bounded gradient. This results in controlled stepsize when doing gradient descent contrarily to KL- and squared Hellinger-GEMINIs. Notice that the change of architecture from a MLP to a LeNet-5 unexpectedly halves the scores for the  $f$ -divergences. We believe this drop is due to the change of notion of neighborhood implied by the network architecture.

#### 4.5 Number of non-empty clusters and architecture

We repeat our previous experiment on the MNIST dataset from Sec. 4.4. We choose this time to get 50 clusters at best for both the MI and the MMD GEMINI and train the models for 100 epochs. We repeat the experiment 20 times per model and plot the resulting scores in figures 7a and 7b. We did not choose to test with the Wasserstein GEMINI because its complexity implies a long training time for 50 clusters, as explained in App. H. We first observe in Fig. 7 that the MMD-GEMINI with linear kernel has a tendency to exploit more

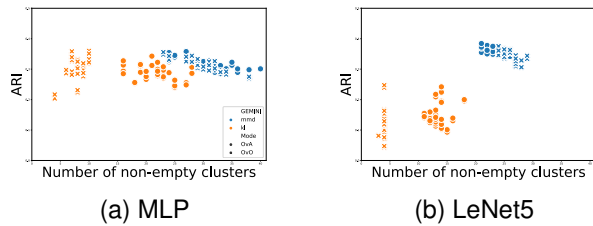


Fig. 7. Distributions of the ARI scores given a number of non-empty clusters after 100 epochs of training on MNIST on two different architectures.

clusters than the MI. The model converges to approximately 30 clusters in the case of the MLP and 25 for the LeNet-5 model with less variance. We can further observe that for all metrics the choice of architecture impacted the number of non-empty clusters after training. Indeed, by playing a key role in the decision boundary shape, the architecture may limit the number of clusters to be found: the MLP can draw more complex boundaries compared to the LeNet5 model. Moreover, we suppose that the cluster selection behaviour of GEMINI may be due to optimisation processes. Indeed, we optimise estimators of the GEMINI rather than the exact GEMINI. Finally, Fig. 7 also confirms from Table. 3 the stability of the MMD-GEMINI regarding the ARI despite the change of architecture whereas the MI is affected and shows poor performance with the LeNet-5 architecture.

#### 4.6 Cifar10 clustering using a SIMCLR-derived kernel

To further illustrate the benefits of the kernel or distance provided to GEMINIs, we continue the same experiment as in section 4.4. However, we focus this time on the CIFAR10 dataset. As improved distance, we chose a linear kernel and  $\ell_2$  norm between features extracted from a pretrained SIMCLR model [13]. We provide results for two different architectures: LeNet-5 and ResNet-18 both trained from scratch on raw images, the latter being a common choice of models in deep clustering literature [61], [64]. We report the results in Table 4 and provide the baseline of MI. We also write the baselines from related works when not using data augmentations to make a fair comparison. Indeed, models trained with GEMINIs do not use data augmentation: only the architecture and the kernel or distance function in the data space plays a role. We observe here that the choice of kernel or distance can be critical in GEMINIs. Indeed, while the Euclidean norm between images does not provide insights on how images of cats and dogs are far as shown by K-Means, features derived from SIMCLR carry much more insight on the proximity of images. This shows that the performances of GEMINIs depend on the quality of distance functions. Interestingly, we observe that for the Resnet-18 using SIMCLR features to guide GEMINIs was not as successful as it has been on the LeNet-5. We believe that the ability of this network to draw any decision boundary makes it equivalent to a categorical distribution model as in Sec. 4.1. Finally, to the best of our knowledge, we are the first to train from scratch a standard discriminative neural network on CIFAR raw images without using labels or direct data augmentations, while getting sensible clustering results.

However, other recent methods achieve best scores using data augmentations which we do not [53].

#### 4.7 A practical application with Graph Neural Networks: the Enron email dataset

We focus in this experiment on the clustering of nodes in a graph representing email interaction between individuals in the Enron dataset (available at <https://www.cs.cmu.edu/~./enron/>). This famous company was filed for bankruptcy on the 2nd of December 2001 following an investigation for fraud by the Securities and Exchange Commission (SEC). Following Bouveyron et al. [7], we focus on exchange of emails in the Enron corporation between the 1st of September 2001 and the 31st of December 2001, which corresponds to the peak period at which the company collapsed. We represent the dataset as a graph where each node corresponds to an employee and each edge represents the sending of at least one email. We choose to let the edges unweighted because the number of emails sent between two persons is not necessarily reflexive of how two persons can be close to each other. After filtering nodes that do not communicate between September and December, the graph comprises 141 nodes and 872 symmetric edges. In order to use the Wasserstein metric, we choose as a distance the all-pairs-shortest path as already used in Section 4.3 and Appendix J. Due to the necessity of using a *symmetric* distance for the Wasserstein metric, we let the graph undirected. Thus, an edge between two persons represent the exchange of at least one email anyhow.

To cluster the nodes of the graph, we adopt a simple Graph Convolution Network (GCN) [39] inspired by Kipf et al. [39] and Liang et al. [44] with a hidden size of dimension 64 and 10 clusters to find at best [7]. We vary the number of hidden layers from 1 to 3. All models were run 30 times. We perform clustering by maximising the OvO Wasserstein during 1000 epochs with a learning rate of  $2 \times 10^{-3}$  for the Adam optimiser. Indeed, with a long training time, we allow the model to find more clusters despite a stabilised GEMINI. We also experimented using the MI with the same models and hyperparameters. We eventually select the final model using the highest GEMINI value for each different depth of GCN.

Similarly, we run 30 times as well two competitors with a number of clusters to find ranging from 2 to 10: the LPM model [32] and the Deep LPM model [44] which are generative methods based on the assumption of a latent position of the nodes in the graph determining their interaction. Their respective best models were selected according to the lowest bayesian information criterion and the highest evidence lower bound. For Deep LPM, we used the architecture proposed by the authors in [44] with a one-hidden-layer network.

We start by showing the highest Wasserstein-GEMINI clustering in Figure 8 where the graph nodes are positioned according to the Fruchterman Reingold algorithm [22].

When we look at the interaction matrices of the best models in Figure 9, we can observe that the GEMINI clustering applied to graph using the all pair shortest path distance yields dense clusters where nodes intensively connect with each others (Figure 9a). The LPM model found

TABLE 4

ARI score of models trained for 200 epochs on CIFAR10 with different architectures using GEMINIs. The kernel for the MMD is either a linear kernel or the dot product between features extracted from a pretrained SIMCLR model. Both the Euclidean norm between images and SIMCLR features are considered for the Wasserstein metric. We report the ARI of related works when not using data augmentation for comparison. \*: scores reported from Li et al., (2021)

Architecture	No kernel	Linear kernel / $\ell_2$ norm				SIMCLR [13]			
	$\mathcal{I}_{\text{KL}}^{\text{ova}}$	$\mathcal{I}_{\text{MMD}}^{\text{ova}}$	$\mathcal{I}_{\text{MMD}}^{\text{ovo}}$	$\mathcal{I}_{\mathcal{W}}^{\text{ova}}$	$\mathcal{I}_{\mathcal{W}}^{\text{ovo}}$	$\mathcal{I}_{\text{MMD}}^{\text{ova}}$	$\mathcal{I}_{\text{MMD}}^{\text{ovo}}$	$\mathcal{I}_{\mathcal{W}}^{\text{ova}}$	$\mathcal{I}_{\mathcal{W}}^{\text{ovo}}$
LeNet-5	0.026	0.049	0.048	0.043	0.041	<b>0.157</b>	0.145	0.079	0.138
Resnet-18	0.008	0.047	0.044	0.037	0.036	0.122	<b>0.145</b>	0.052	0.080
KMeans (images / SIMCLR)			0.041	0.147	CC [43]			0.030	
IDFD [61]			0.060		JULE [69]*			0.138	

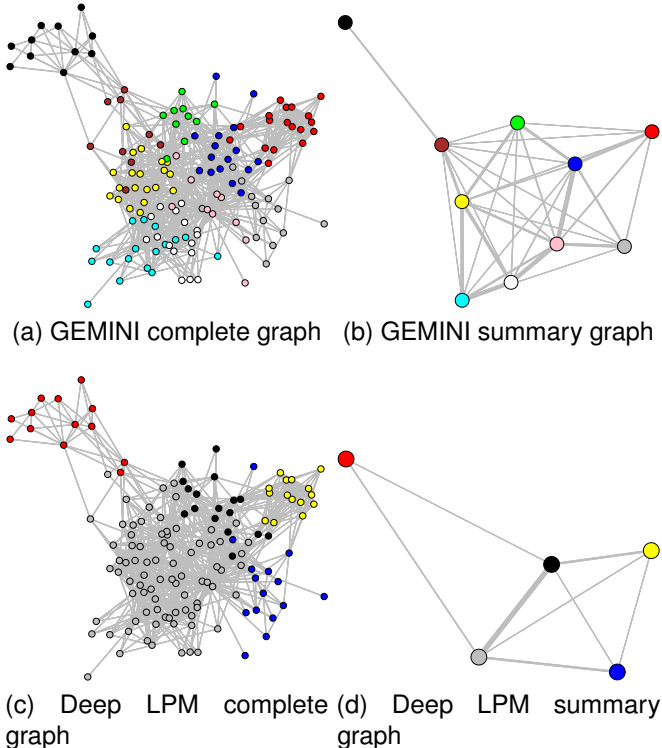


Fig. 8. Fruchterman Reingold representation of the Enron email interaction graph. Nodes are coloured according to clusters. The summary graph is a cluster-wise average of the positions of the nodes with edges as strong as the number of interactions between two clusters.

TABLE 5

Average ARI scores between 30 GEMINI-trained models and the best Deep LPM and LPM clustering according to ELBO criterion.  $H$  represents the number of hidden layers in the trained model.

GEMINI	$H$	LPM	Deep LPM
MI	1	0.47 (0.08)	<b>0.31 (0.10)</b>
	2	0.31 (0.07)	0.17 (0.09)
	3	0.29 (0.04)	0.16 (0.05)
OvO Wasserstein	1	<b>0.58 (0.04)</b>	0.25 (0.03)
	2	0.54 (0.06)	0.22 (0.03)
	3	0.49 (0.06)	0.21 (0.04)

fewer clusters that are as well densely connected while the Deep LPM found 6 clusters including one densely connected and another one which contains multiple nodes sparsely connected. Interestingly, GEMINI managed to isolate graph nodes that act as hub e.g. the 9th cluster in Fig. 9a with

star-shaped interactions while Deep LPM mixes hubs in its 3rd cluster in Figure 9c. A potential explanation for this difference is that LPM models seek to cluster nodes based on supposedly close latent representations whereas the GEMINI clustering uses the Wasserstein distance, hence taking into account the flow of information passing through each graph node. In Table 5, we compared the average ARI between our 30 GEMINI models with the best selected LPM and Deep LPM clustering. It appears indeed that GEMINI models have a stronger ARI with the LPM model than with the Deep LPM, as the former concentrated on dense clusters. Interestingly, we can notice in Table 5 that the MI models with one hidden layer are able to produce also satisfying results with an ARI close to or better than the ARI between GEMINI and LPM methods. However, when the models turn deeper, the performances of MI drop whereas the ARI between the Wasserstein-GEMINI decreases more slowly.

To further compare the performances of these models, we chose to evaluate their proportions of ambiguous clusters (PAC score) [71]. The PAC score is evaluated using two steps: we first build a matrix where each entry  $i, j$  contains the proportion of times two samples  $i$  and  $j$  where clustered together by some model, then arrange these proportions to build a cdf which highlights for given quantiles the proportions of paired samples that can be clustered together. The PAC score corresponds to the difference of cdf between two given quantiles, commonly 10% and 90%. Through Figure 10, we can see that the models trained with the Wasserstein GEMINI have a lower PAC score compared to MI-trained models which highlights a more consistent clustering assignment through all 30 runs. We can further observe that for both objective functions, a deeper model leads to higher PAC score which we can explain by the difficulty to repeat the same decision boundary with deep models. Eventually, we may conclude that while the MI may have competitive results with shallow models compared to GEMINI and Deep LPM, it is unable to repeat a consistent clustering as illustrated by a high PAC score.

## 5 CONCLUSIONS

We highlighted that the choice of distance at the core of MI can alter the performances of deep learning models when used as an objective for a deep discriminative clustering. We first showed that MI maximisation does not necessarily reflect the best decision boundary in clustering when the clustering model converges to a Dirac distribution. We introduced the GEMINI, a method which only needs the



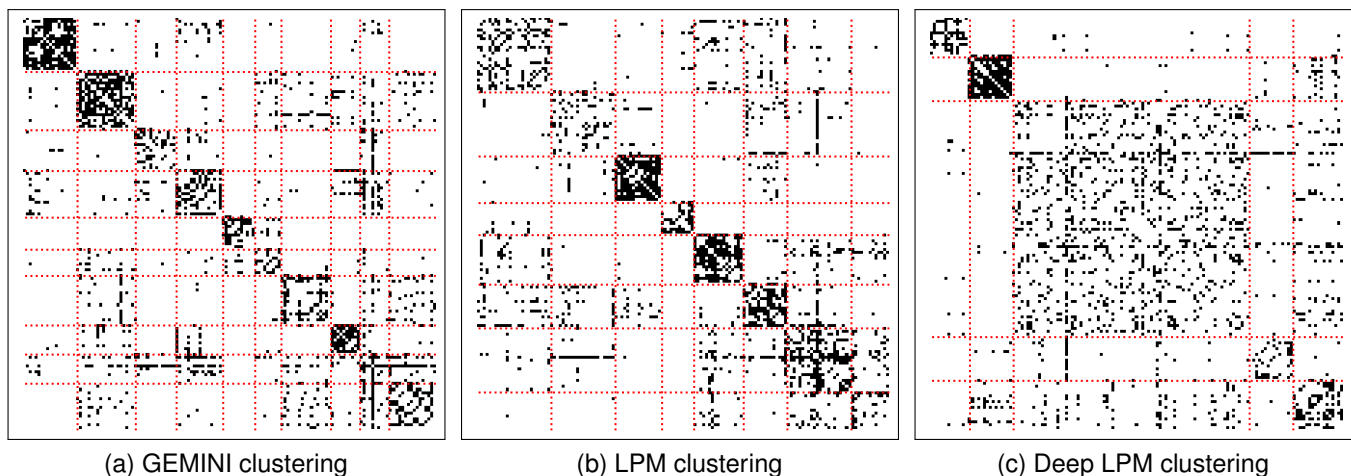


Fig. 9. Interaction matrices: each black cell encodes an edge. Nodes were reorganised according to their clustering for each model.

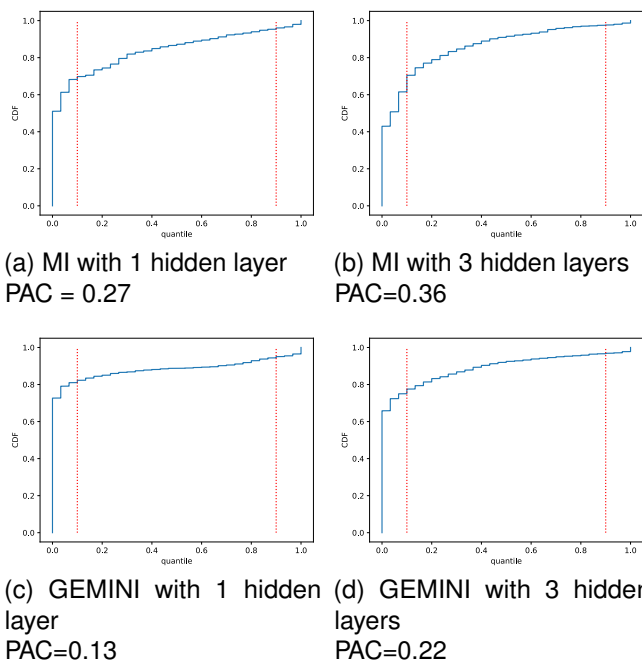


Fig. 10. CDFs of consensus between all clusterings produced by the models on the Enron Email dataset. The indicated PAC score is the difference of cdf between the quantiles 10% and 90%. A lower PAC score is better and highlights more consistent clustering assignments.

specification of a neural network and a kernel or distance in the data space. Moreover, we showed how the notion of neighborhood built by the neural network can affect the clustering, especially for MI. To the best of our knowledge, this is the first method that trains single-stage neural networks from scratch using neither data augmentations nor regularisations, yet achieving good clustering performances. We emphasised that GEMINIs are only searching for a maximum number of clusters: after convergence some may be empty. However, we do not have insights to explain this convergence which is part of future work. Finally, we introduced several versions of GEMINIs and would encourage the OvA MMD or OvA Wasserstein as a default choice, since it proves to both incorporate knowledge from the data using a kernel or

distance while remaining the less complex than OvO MMD and OvO Wasserstein in time and memory. OvO versions could be privileged for fine-tuning steps. As follow-up, we focused on the introduction of sparsity in GEMINI clustering for feature selection [51]. Future works could focus on the joint learning of distances or kernels [67] while maximising the the GEMINI to get both meaningful clustering and metrics in the data space. We will also investigate why sometimes after convergence some clusters end up empty.

#### Acknowledgements

This work has been supported by the French government, through the 3IA Côte d’Azur, Investment in the Future, project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002. We would also like to thank the France Canada Research Fund (FFCR) for their contribution to the project. This work was partly supported by the Fonds de recherche du Québec – Santé (FRQS) and the Health-Data Hub, through the joint project AORTIC STENOSIS. The authors are grateful to the OPAL infrastructure from Université Côte d’Azur for providing resources and support.

#### REFERENCES

- [1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein Generative Adversarial Networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, Aug. 2017.
- [2] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and Equilibrium in Generative Adversarial Nets (GANs). In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 224–232. PMLR, 06–11 Aug 2017.
- [3] D. Barber and F. V. Agakov. The im algorithm: A variational approach to information maximization. In *NIPS*, 2003.
- [4] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm. Mutual Information Neural Estimation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 531–540. PMLR, 10–15 Jul 2018.
- [5] P. Billingsley. *Convergence of Probability Measures*. John Wiley & Sons, 2013.

- [6] C. Bouveyron, G. Celeux, T. B. Murphy, and A. E. Raftery. *Model-based Clustering and Classification for Data Science: with applications in R*. in Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.
- [7] C. Bouveyron, P. Latouche, and R. Zreik. The stochastic topic block model for the clustering of vertices in networks with textual edges. *Statistics and Computing*, 28(1):11–31, 2018.
- [8] J. Bridle, A. Heading, and D. MacKay. Unsupervised Classifiers, Mutual Information and Phantom Targets. In J. Moody, S. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1992.
- [9] U. Caglar. *Divergence And Entropy Inequalities For Log Concave Functions*. PhD thesis, Case Western Reserve University, 2014.
- [10] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018.
- [11] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc., 2020.
- [12] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep Adaptive Image Clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888, 2017.
- [13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.
- [14] A. Corduneanu and T. Jaakkola. On Information Regularization. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 151–158, 2002.
- [15] I. Csizsár. Information-Type Measures of Difference of Probability Distributions and Indirect Observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.
- [16] Z. Dang, C. Deng, X. Yang, K. Wei, and H. Huang. Nearest Neighbor Matching for Deep Clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13693–13702, 2021.
- [17] K. Do, T. Tran, and S. Venkatesh. Clustering by Maximizing Mutual Information Across Views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9928–9938, 2021.
- [18] M. D. Donsker and S. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on pure and applied mathematics*, 36(2):183–212, 1983.
- [19] W. Falcon et al. Pytorch Lightning. *GitHub. Note: <https://github.com/PyTorchLightning/pytorch-lightning>*, 3, 2019.
- [20] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Yayer. Pot: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.
- [21] G. França, M. L. Rizzo, and J. T. Vogelstein. Kernel k-groups via hartigan’s method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4411–4425, 2020.
- [22] T. M. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.
- [23] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5736–5745, 2017.
- [24] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.
- [25] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. Regularisation of Neural Networks by Enforcing Lipschitz Continuity. *Machine Learning*, 110(2):393–416, Feb. 2021.
- [26] R. M. Gray and P. C. Shields. The Maximum Mutual Information between Two Random Processes. *Information and Control*, 33(4):273–280, 1977.
- [27] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A Kernel Two-Sample Test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. Publisher: JMLR. org.
- [28] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.
- [29] W. Harchaoui. *Learning representations using neural networks and optimal transport*. PhD thesis, Université Paris Cité, 2020.
- [30] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.
- [31] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.
- [32] P. D. Hoff, A. E. Raftery, and M. S. Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.
- [33] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning Discrete Representations via Information Maximizing Self-Augmented Training. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1558–1567. PMLR, Aug. 2017.
- [34] J. Huang, S. Gong, and X. Zhu. Deep Semantic Clustering by Partition Confidence Maximisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8849–8858, 2020.
- [35] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.
- [36] M. Jabi, M. Pedersoli, A. Mitiche, and I. B. Ayed. Deep clustering: On the link between discriminative models and K-means. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.
- [37] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant Information Clustering for Unsupervised Image Classification and Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.
- [38] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [39] T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.
- [40] J. Kleinberg. An Impossibility Theorem for Clustering. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2003.
- [41] A. Krause, P. Perona, and R. Gomes. Discriminative Clustering by Regularized Information Maximization. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.
- [42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [43] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng. Contrastive Clustering. In *2021 AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- [44] D. Liang, M. Corneli, C. Bouveyron, and P. Latouche. Clustering by deep latent position model with graph convolutional network. 2022.
- [45] F. Liese and I. Vajda. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.
- [46] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long. A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture. *IEEE Access*, PP:1–1, 2018.
- [47] T. Minka. Discriminative models, not discriminative training. Technical report, Technical Report MSR-TR-2005-144, Microsoft Research, 2005.
- [48] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations*, 2018.
- [49] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018. Publisher: IEEE.



- [50] X. Nguyen, M. J. Wainwright, and M. I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.
- [51] L. Ohl, P.-A. Mattei, C. Bouveyron, M. Leclercq, A. Droit, and F. Precioso. Sparse gemini for joint discriminative clustering and feature selection. *arXiv preprint arXiv:2302.03391*, 2023.
- [52] A. B. Owen. *Monte Carlo theory, methods and examples*. Stanford, 2013.
- [53] S. Park, S. Han, S. Kim, D. Kim, S. Park, S. Hong, and M. Cha. Improving Unsupervised Image Clustering With Robust Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12278–12287, 2021.
- [54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [56] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On Variational Bounds of Mutual Information. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5171–5180. PMLR, June 2019.
- [57] A. Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4, pages 547–562. University of California Press, 1961.
- [58] M. Ronen, S. E. Finder, and O. Freifeld. DeepDPM: Deep Clustering With an Unknown Number of Clusters. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9851–9860, 2022.
- [59] B. Roy. Transitivity et connexité. *Comptes Rendus Hebdomadaires Des Séances De L'Académie Des Sciences*, 249(2):216–218, 1959.
- [60] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. Lanckriet. On integral probability metrics,  $\phi$ -divergences and binary classification. *arXiv preprint arXiv:0901.2698*, 2009.
- [61] Y. Tao, K. Takagi, and K. Nakata. Clustering-friendly representation learning via instance discrimination and feature decorrelation. In *International Conference on Learning Representations*, 2021.
- [62] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic. On Mutual Information Maximization for Representation Learning. In *International Conference on Learning Representations*, 2020.
- [63] A. Van den Oord, Y. Li, and O. Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv e-prints*, pages arXiv-1807, 2018.
- [64] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool. Scan: Learning to Classify Images without Labels. In *European Conference on Computer Vision*, pages 268–285. Springer, 2020.
- [65] C. Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.
- [66] S. Warshall. A Theorem on Boolean Matrices. *Journal of the ACM (JACM)*, 9(1):11–12, 1962.
- [67] C. Wu, Z. Khan, S. Ioannidis, and J. G. Dy. Deep kernel learning for clustering. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 640–648. SIAM, 2020.
- [68] J. Xie, R. Girshick, and A. Farhadi. Unsupervised Deep Embedding for Clustering Analysis. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 478–487, New York, New York, USA, June 2016. PMLR.
- [69] J. Yang, D. Parikh, and D. Batra. Joint Unsupervised Learning of Deep Representations and Image Clusters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016.
- [70] Z. Zhang and Y. Boykov. Revisiting discriminative entropy clustering and its relation to k-means. *arXiv preprint arXiv:2301.11405*, 2023.
- [71] Y. Şenbabaoğlu, G. Michailidis, and J. Z. Li. Critical limitations of consensus clustering in class discovery. *Scientific reports*, 4(1):6207, 2014.



**Louis Ohl** PhD Student at the Université Côte d'Azur and the Université Laval from 2021 to 2024 and computer science graduate from the INSA Lyon and KTH Royal Institute of Technology in 2021. His project mainly focuses on clustering and the choices of the decision boundaries to discriminatively separate samples with applications in medical research including identification of phenogroups in aortic stenosis.



**Pierre-Alexandre Mattei** is a research scientist at Inria, in the Maasai team located in Sophia-Antipolis (France). He graduated from Ecole Normale Supérieure de Cachan in 2014, and received a PhD in applied mathematics from Université Paris Cité in 2017. His research interests mostly revolve around problems with latent variables: deep generative modelling, Bayesian inference, clustering, missing data.

**Charles Bouveyron** Professor of Statistics at Université Côte d'Azur, Nice, France. He holds a Chair on Artificial Intelligence and is the Director of the Institut 3IA Côte d'Azur and head of the research team MAASAI from INRIA. His research interests include statistical learning for clustering, classification and regression in high dimension, statistical learning on networks applied to different domains (medicine, image analysis, astrophysics, humanities...)



**Warith Harchaoui** holds a Ph.D. in Applied Mathematics and is deeply engaged in the intersection of artificial intelligence (AI) and pattern analysis. He commenced his academic journey at École Normale Supérieure de Cachan, where he received his M.Sc. in 2008. His initial forays into the industry revolved around Computer Vision applications in startups and global enterprises. Harchaoui extended his academic pursuits at École Normale Supérieure de Paris, contributing to research at the globally recognized Willow laboratory. His scholarly interests encompass a range of topics within image, sound, video, and text processing, a domain where he has consistently leveraged mathematical models for innovative solutions. From 2014 to 2020, Dr. Harchaoui also embraced a corporate role in Data Science, working with Oscaro.com the e-commerce leader in the automotive parts sector. Balancing industry demands with academic rigor, he completed his Ph.D. research between 2016 and 2020. Today, Dr. Harchaoui serves at Jellysmack since 2021, where he combines his expertise to help internet video creators through every stage from content production, editing to social media promotion.



**Mickaël Leclercq** PhD in Computer Science from McGill University, Canada, in 2016. Since then, he's been working as a research associate in the computational biology laboratory of Prof. Droit, researcher at the CHU de Québec - Université Laval. His role is to develop and supervise bioinformatics research projects with an artificial intelligence component. He also contributes to the organization and coordination of the laboratory's research activities and supervises the work of students and young research professionals.

He actively participates in the long-term vision of the laboratory's research themes as well as in the development of grant applications and manuscripts.



**Arnaud Droit** PhD graduate in 2007 in bioinformatics from Université Laval, Canada. Since 2012, he has been a Professor at Université Laval (Québec, Canada) and leads a bioinformatics and proteomics laboratory at the CHU de Québec. His research focuses on robust bioinformatics approaches, advanced statistical methods, and machine/deep learning strategies to extract information relevant to medical research from large biological datasets. Many of his projects are related to cancer research, where the aim is to discover

biomarker signatures associated with various clinical outcomes. His expertise is essential to many research sectors requiring bioinformatics, so he built a large network of international collaborations, including the private sector.

**Frederic Precioso** Full Professor at the Université Côte d'Azur from 2011 and Affiliated Professor at Université Laval from 2022, his research interests cover active learning, foundation models, hybrid learning combining symbolic and non-symbolic models, clustering, MLOps, applied to different domains (health, biology, multimedia, autonomous vehicles, biodiversity).

## APPENDIX A

### DEMONSTRATION OF THE CONVERGENCE TO 0 OF THE MI FOR A GAUSSIAN MIXTURE

#### A.1 Models definition

Let us consider a mixture of two Gaussian distributions, both with different means  $\mu_0$  and  $\mu_1$ , s.t.  $\mu_0 < \mu_1$  and of same standard deviation  $\sigma$ :

$$p(x|y=0) = \mathcal{N}(x|\mu_0, \sigma^2), p(x|y=1) = \mathcal{N}(x|\mu_1, \sigma^2),$$

where  $y$  is the cluster assignment. We take balanced clusters proportions, i.e.  $p(y=0) = p(y=1) = \frac{1}{2}$ . This first model is the basis that generated the complete dataset  $p(x)$ . When performing clustering with our discriminative model, we are not aware of the distribution. Consequently: we create other models. We want to compute the difference of mutual information between two decision boundaries that discriminative models  $p_\theta(y|x)$  may yield.

We define two decision boundaries: one which splits evenly the data space called  $p_A$  and another which splits it on a closed set  $p_B$ :

$$p_A(y=1|x) = \begin{cases} 1 - \epsilon & x > \frac{\mu_1 - \mu_0}{2} \\ \epsilon & \text{otherwise} \end{cases}, \quad (16)$$

$$p_B(y=1|x) = \begin{cases} 1 - \epsilon & x \in [\mu_0, \mu_1] \\ \epsilon & \text{otherwise} \end{cases}. \quad (17)$$

Our goal is to show that both models  $p_A$  and  $p_B$  will converge to the same value of mutual information as  $\epsilon$  converges to 0.

#### A.2 Computing cluster proportions

##### A.2.1 Cluster proportion of the correct decision boundary

To compute the cluster proportions, we estimate with samples  $x$  from the distribution  $p_{\text{data}}(x)$ . Since we are aware for this demonstration of the true nature of the data distribution, we can use  $p(x)$  for sampling. Consequently, we can compute the two marginals:

$$\begin{aligned} p_A(y=1) &= \int_{\mathcal{X}} p(x)p_A(y=1|x)dx, \\ &= \int_{-\infty}^{\frac{\mu_1 - \mu_0}{2}} p(x)\epsilon dx + \int_{\frac{\mu_1 - \mu_0}{2}}^{+\infty} p(x)(1 - \epsilon)dx, \\ &= \epsilon \left( \int_{-\infty}^{\frac{\mu_1 - \mu_0}{2}} p(x)dx \right) + (1 - \epsilon) \left( \int_{\frac{\mu_1 - \mu_0}{2}}^{+\infty} p(x)dx \right), \\ &= \frac{1}{2}. \end{aligned}$$

##### A.2.2 Cluster proportion of the misplaced decision boundary

For the misplaced decision boundary, the marginal is different:

$$\begin{aligned} p_B(y=1) &= \int_{\mathcal{X}} p(x)p_B(y=1|x)dx, \\ &= \epsilon \left( \int_{-\infty}^{\mu_0} p(x)dx + \int_{\mu_1}^{+\infty} p(x)dx \right) + (1 - \epsilon) \int_{\mu_0}^{\mu_1} p(x)dx, \\ &= \epsilon \left( 1 - \int_{\mu_0}^{\mu_1} p(x)dx \right) + (1 - \epsilon) \int_{\mu_0}^{\mu_1} p(x)dx. \end{aligned}$$

Here, we simply introduce a new variable named  $\beta$  that will be a shortcut for noting the proportion of data between  $\mu_0$  and  $\mu_1$ :

$$\beta = \int_{\mu_0}^{\mu_1} p(x)dx.$$

And so can we simply write the cluster proportion of decision boundary model B as:

$$p_B(y=1) = \epsilon(1 - \beta) + (1 - \epsilon)\beta,$$

Leading to the summary of proportions in Table 6. For convenience, we will write the proportions of model B using the shortcuts:

$$\begin{aligned}\pi_B &= p_B(y = 1) = \epsilon + \beta(1 - 2\epsilon), \\ \bar{\pi}_B &= p_B(y = 0) = 1 - \epsilon - \beta(1 - 2\epsilon).\end{aligned}$$

TABLE 6  
Proportions of clusters for models A and B

$\mathcal{M}$	A	B
$p_{\mathcal{M}}(y = 1)$	$\frac{1}{2}$	$\epsilon + \beta(1 - 2\epsilon)$
$p_{\mathcal{M}}(y = 0)$	$\frac{1}{2}$	$1 - \epsilon - \beta(1 - 2\epsilon)$

### A.3 Computing the KL divergences

#### A.3.1 Correct decision boundary

We first start by computing the Kullback-Leibler divergence for some arbitrary value of  $x \in \mathbb{R}$ :

$$D_{\text{KL}}(p_A(y|x)||p_A(y)) = \sum_{i=0}^1 p_A(y = i|x) \log \frac{p_A(y = i|x)}{p_A(y = i)}.$$

We now need to detail the specific cases, for the value of  $p(y = i|x)$  is dependent on  $x$ . We start  $\forall x < \frac{\mu_1 - \mu_0}{2}$ :

$$\begin{aligned}D_{\text{KL}}(p_A(y|x)||p_A(y)) &= p_A(y = 0|x) \log \frac{p_A(y = 0|x)}{\frac{1}{2}} + p_A(y = 1|x) \log \frac{p_A(y = 1|x)}{\frac{1}{2}}, \\ &= (1 - \epsilon) \log 2(1 - \epsilon) + \epsilon \log 2\epsilon.\end{aligned}$$

The opposite case,  $\forall x \geq \frac{\mu_1 - \mu_0}{2}$  yields:

$$\begin{aligned}D_{\text{KL}}(p_A(y|x)||p_A(y)) &= p_A(y = 0|x) \log \frac{p_A(y = 0|x)}{\frac{1}{2}} + p_A(y = 1|x) \log \frac{p_A(y = 1|x)}{\frac{1}{2}}, \\ &= \epsilon \log 2\epsilon + (1 - \epsilon) \log 2(1 - \epsilon).\end{aligned}$$

Since both cases are equal, we can write down:

$$D_{\text{KL}}(p_A(y|x)||p_A(y)) = \epsilon \log 2\epsilon + (1 - \epsilon) \log 2(1 - \epsilon), \forall x \in \mathbb{R}. \quad (18)$$

#### A.3.2 Misplaced boundary

We proceed to the same detailing of the Kullback-Leibler divergence computation for the misplaced decision boundary. We start with the set  $x \in [\mu_0, \mu_1]$ :

$$\begin{aligned}D_{\text{KL}}(p_B(y|x)||p_B(y)) &= p_B(y = 0|x) \log \frac{p_B(y = 0|x)}{p_B(y = 0)} + p_B(y = 1|x) \log \frac{p_B(y = 1|x)}{p_B(y = 1)}, \\ &= \epsilon \log \frac{\epsilon}{\bar{\pi}_B} + (1 - \epsilon) \log \frac{1 - \epsilon}{\pi_B}.\end{aligned}$$

When  $x$  is out of this set, the divergence becomes:

$$\begin{aligned}D_{\text{KL}}(p_B(y|x)||p_B(y)) &= p_B(y = 0|x) \log \frac{p_B(y = 0|x)}{p_B(y = 0)} + p_B(y = 1|x) \log \frac{p_B(y = 1|x)}{p_B(y = 1)}, \\ &= (1 - \epsilon) \log \frac{1 - \epsilon}{\bar{\pi}_B} + \epsilon \log \frac{\epsilon}{\pi_B}.\end{aligned}$$

To fuse the two results, we will write the KL divergence as such:

$$D_{\text{KL}}(p_B(y|x)||p_B(y)) = \epsilon \log \epsilon + (1 - \epsilon) \log(1 - \epsilon) - C(x), \forall x \in \mathbb{R}, \quad (19)$$

where  $C(x)$  is a constant term depending on  $x$  defined by:

$$C(x) = \begin{cases} \epsilon \log \bar{\pi}_B + (1 - \epsilon) \log \pi_B & x \in [\mu_0, \mu_1] \\ \epsilon \log \pi_B + (1 - \epsilon) \log \bar{\pi}_B & x \in \mathbb{R} \setminus [\mu_0, \mu_1] \end{cases}.$$

For simplicity of later writings, we will shorten the notations by:

$$C(x) = \begin{cases} \alpha_1 & x \in [\mu_0, \mu_1] \\ \alpha_0 & x \in \mathbb{R} \setminus [\mu_0, \mu_1] \end{cases}.$$

## A.4 Evaluating the mutual information

### A.4.1 Correct decision boundary

We inject the value of the Kullback-Leibler divergence from Eq. (18) inside an expectation performed over the data distribution  $p_{\text{data}}(x)$ :

$$\mathcal{I}_A(x; y) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [D_{\text{KL}}(p_A(y|x) || p_A(y))], \quad (20)$$

$$= \int_{\mathcal{X}} p(x) (\epsilon \log(2\epsilon) + (1 - \epsilon) \log(2(1 - \epsilon))) dx, \quad (21)$$

$$= \epsilon \log(2\epsilon) + (1 - \epsilon) \log(2(1 - \epsilon)). \quad (22)$$

Since the KL divergence was independent of  $x$ , we could leave the constant outside of the integral which is equal to 1.

We can assess the coherence of Eq. (22) since its limit as  $\epsilon$  approaches 0 is  $\log 2$ . In terms of bits, this is the same as saying that the information on  $X$  directly gives us information on the  $Y$  of the cluster.

### A.4.2 Odd decision boundary

We inject the value of the KL divergence from Eq. (19) inside the expectation of the mutual information:

$$\begin{aligned} \mathcal{I}_B(x; y) &= \mathbb{E}_{x \sim p_{\text{data}}(x)} [D_{\text{KL}}(p_B(y|x) || p_B(y))], \\ &= \int_{\mathcal{X}} p(x) (\epsilon \log \epsilon + (1 - \epsilon) \log(1 - \epsilon) - C(x)) dx, \\ &= \epsilon \log \epsilon + (1 - \epsilon) \log(1 - \epsilon) - \int_{\mathcal{X}} p(x) C(x) dx. \end{aligned}$$

The first terms are constant with respect to  $x$  and the integral of  $p(x)$  over  $\mathcal{X}$  adds up to 1. We finally need to detail the expectation of the constant  $C(x)$  from Eq. (A.3.2):

$$\begin{aligned} \mathbb{E}_x[C(x)] &= \int_{-\infty}^{\mu_0} C(x)p(x)dx + \int_{\mu_0}^{\mu_1} C(x)p(x)dx + \int_{\mu_1}^{+\infty} C(x)p(x)dx, \\ &= \alpha_0 \left( \int_{-\infty}^{\mu_0} p(x)dx + \int_{\mu_1}^{+\infty} p(x)dx \right) + \alpha_1 \int_{\mu_0}^{\mu_1} p(x)dx, \\ &= \alpha_0(1 - \beta) + \alpha_1\beta. \end{aligned}$$

This can be further improved by unfolding the description of  $\alpha_0$  and  $\alpha_1$  from Eq. (A.3.2):

$$\begin{aligned} \alpha_0(1 - \beta) + \beta\alpha_1 &= \alpha_0 + \beta(\alpha_1 - \alpha_0), \\ &= \epsilon \log \pi_B + (1 - \epsilon) \log \bar{\pi}_B + \beta [\epsilon \log \bar{\pi}_B + (1 - \epsilon) \log \pi_B \\ &\quad - \epsilon \log \pi_B - (1 - \epsilon) \log \bar{\pi}_B], \\ &= [1 - \epsilon + \beta\epsilon - \beta + \beta\epsilon] \log \bar{\pi}_B + [\epsilon + \beta - \beta\epsilon - \beta\epsilon] \log \pi_B, \\ &= \log \bar{\pi}_B + [2\beta\epsilon - \beta - \epsilon] \log \frac{\bar{\pi}_B}{\pi_B}. \end{aligned}$$

We can finally write down the mutual information for the model B:

$$\mathcal{I}_B(x; y) = \epsilon \log \epsilon + (1 - \epsilon) \log(1 - \epsilon) - \log \bar{\pi}_B - [2\beta\epsilon - \beta - \epsilon] \log \frac{\bar{\pi}_B}{\pi_B}.$$

## A.5 Differences of mutual information

Now that we have the exact value of both mutual informations, we can compute their differences:

$$\begin{aligned}\Delta_{\mathcal{I}} &= \mathcal{I}_A(x; y) - \mathcal{I}_B(x; y), \\ &= \epsilon \log(2\epsilon) + (1 - \epsilon) \log(2(1 - \epsilon)) - \epsilon \log \epsilon - (1 - \epsilon) \log(1 - \epsilon) \\ &\quad + \log \bar{\pi}_B + [2\beta\epsilon - \beta - \epsilon] \log \frac{\bar{\pi}_B}{\pi_B}, \\ &= \epsilon \log 2 + (1 - \epsilon) \log 2 + \log \bar{\pi}_B + [2\beta\epsilon - \beta - \epsilon] \log \frac{\bar{\pi}_B}{\pi_B}.\end{aligned}$$

We then deduce how the difference of mutual information evolves as the decision boundary becomes sharper, i.e. when  $\epsilon$  approaches 0:

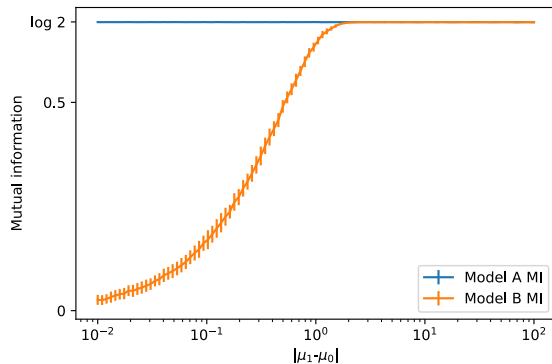
$$\lim_{\epsilon \rightarrow 0} \Delta_{\mathcal{I}} = \log 2 + \log \bar{\pi}_B - \beta \log \frac{\bar{\pi}_B}{\pi_B}.$$

However, the cluster proportions by B  $\pi_B$  also take a different value as  $\epsilon$  approaches 0. Recalling Eq. (17):

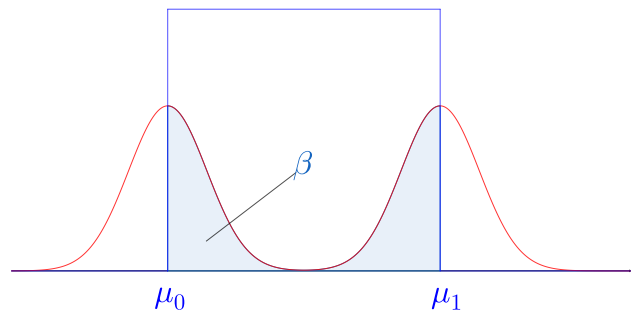
$$\lim_{\epsilon \rightarrow 0} p_B(y = 1) = \beta, \quad \lim_{\epsilon \rightarrow 0} p_B(y = 0) = 1 - \beta.$$

And finally can we write that:

$$\begin{aligned}\lim_{\epsilon \rightarrow 0} \Delta_{\mathcal{I}} &= \log 2 + \log(1 - \beta) - \beta \log \frac{1 - \beta}{\beta}, \\ &= \log 2 + (1 - \beta) \log(1 - \beta) + \beta \log \beta, \\ &= \log 2 - \mathcal{H}(\beta).\end{aligned}$$



(a) Differences of MI between models A and B



(b) Gaussian mixture distribution  $p(x)$  with proportion  $\beta$  in between the two means  $\mu_0$  and  $\mu_1$

Fig. 11. Value of the mutual information for the two models splitting a Gaussian mixture depending on the distance between the two means  $\mu_0$  and  $\mu_1$  of the two generating Gaussian distributions. We estimate the MI by computing it 50 times on 1000 samples drawn from the Gaussian mixture.

To conclude, as the decision boundaries turn sharper, i.e. when  $\epsilon$  approaches 0, the difference of mutual information between the two models is controlled by the entropy of proportion of data  $\beta$  between the two means  $\mu_0$  and  $\mu_1$ . We know that for binary entropies, the optimum is reached for  $\beta = 0.5$ . In other words having  $\mu_0$  and  $\mu_1$  distant enough to ensure balance of proportions between the two clusters of model B leads to a difference of mutual information equal to 0. We experimentally highlight this convergence in Figure 11 where we compute the mutual information of models A and B as the distance between the two means  $\mu_0$  and  $\mu_1$  increases in the Gaussian distribution mixture.

## APPENDIX B

### PROOF OF PROPOSITION 3.1

For the sake of clarity, we will use the notations  $\pi_k \equiv p(y = k)$  during the demonstration.



### B.1 Modelisation of the conditional distribution

We will consider two types of models. The first one is the *generic* clustering model, where the cluster assignment follows a categorical distribution:

$$y|\mathbf{x} \sim \text{Cat}(\psi_\theta(\mathbf{x})),$$

where  $\psi_\theta : \mathcal{X} \rightarrow \Delta_K$  is a learnable function of parameters  $\theta$  and  $\Delta_K$  is a  $K$ -simplex.

The second model is a Dirac distribution where the data space  $\mathcal{X}$  is divided into a partition  $\mathcal{X}_k, \forall k \in \{1, \dots, K\}$ :

$$p(y = k|\mathbf{x}) = \mathbf{1}_{[\mathbf{x} \in \mathcal{X}_k]},$$

with  $\mathbf{1}$  the indicator function. This is simply a sub-case of the generic model.

For both models, we consider that clusters are not empty and that the model is not degenerate, i.e.  $\pi_k \in ]0, 1[ \forall k \in \{1, K\}$ .

### B.2 Value of the GEMINI OvA and upper bounds

We first unfold the OvA GEMINI for the generic model and  $\alpha \in \mathbb{R} \setminus \{0, 1\}$ :

$$\begin{aligned} \mathcal{I}_{D_\alpha}^{\text{OvA}}(\mathbf{x}; y) &= \mathbb{E}_{y \sim p(y)} \left[ \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ f_\alpha \left( \frac{p(\mathbf{x}|y)}{p(\mathbf{x})} \right) \right] \right], \\ &= \sum_{k=1}^K \pi_k \int_{\mathcal{X}} p(\mathbf{x}) \left( \frac{p(\mathbf{x}|y=k)^\alpha p(\mathbf{x})^{-\alpha}}{\alpha(\alpha-1)} - \frac{p(\mathbf{x}|y=k)p(\mathbf{x})^{-1}}{\alpha-1} + \frac{1}{\alpha} \right) d\mathbf{x}, \\ &= \sum_{k=1}^K \pi_k \int_{\mathcal{X}} \left( \frac{p(\mathbf{x})p(y=k|\mathbf{x})^\alpha}{\pi_k^\alpha \alpha(\alpha-1)} - \frac{p(\mathbf{x}|y=k)}{\alpha-1} + \frac{p(\mathbf{x})}{\alpha} \right) d\mathbf{x}. \end{aligned}$$

After distributing the factor  $p(\mathbf{x})$  in the integral, we can notice that the two last terms will be summed to 1, up to a factor depending on  $\alpha$ .

$$\begin{aligned} \mathcal{I}_{D_\alpha}^{\text{OvA}}(\mathbf{x}; y) &= \sum_{k=1}^K \pi_k \left( \frac{1}{\alpha} - \frac{1}{\alpha-1} + \frac{1}{\pi_k^\alpha \alpha(\alpha-1)} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y=k|\mathbf{x})^\alpha] \right) \\ &= \frac{-1}{\alpha(\alpha-1)} + \frac{1}{\alpha(\alpha-1)} \sum_{k=1}^K \pi_k^{1-\alpha} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y=k|\mathbf{x})^\alpha], \\ &= (\alpha(\alpha-1))^\alpha \left[ -1 + \sum_{k=1}^K \pi_k^{1-\alpha} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y=k|\mathbf{x})^\alpha] \right]. \end{aligned}$$

Since we face a categorical distribution, we can affirm that  $p(y|\mathbf{x}) \in [0, 1]$ . Therefore, depending on the value of  $\alpha$ , we have either  $p(y=k|\mathbf{x})^\alpha \in [1, \infty[$  if  $\alpha$  is negative, and  $p(y=k|\mathbf{x}) \in [0, 1]$  for  $\alpha$  positive. We now inspect these different cases.

### B.3 Case of $\alpha \in ]1, +\infty[$

The upper bound we can get on the expectation involves the inequality  $p(y=k|\mathbf{x})^\alpha \leq p(y=k|\mathbf{x})$ . Owing to the linearity of the expectation, we can affirm that:

$$\begin{aligned} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y=k|\mathbf{x})^\alpha] &\leq \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y=k|\mathbf{x})], \\ &\leq \pi_k. \end{aligned}$$

This allows us to derive the following upper bound on the OvA GEMINI:

$$\begin{aligned} \mathcal{I}_{D_\alpha}^{\text{OvA}}(\mathbf{x}; y) &= (\alpha(\alpha-1))^{-1} \left[ -1 + \sum_{k=1}^K \pi_k^{1-\alpha} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y=k|\mathbf{x})^\alpha] \right], \\ &\leq (\alpha(\alpha-1))^{-1} \left[ -1 + \sum_{k=1}^K \pi_k^{2-\alpha} \right], \\ &\leq \mathcal{B}_{]1, +\infty[}(\pi_1, \dots, \pi_K). \end{aligned}$$

The upper bound  $\mathcal{B}_{]1, +\infty[}(\pi_1, \dots, \pi_K)$  is a convex function that is invariant to permutations of  $\pi_k$ . Interestingly, this upper bound only depends on the proportions of the clusters. Its maximum is reached when  $\pi_k = K^{-1}$ . However, any

solution is acceptable for the special case of  $\alpha = 2$ , i.e. the Pearson  $\chi^2$ -divergence. In this case, the upper bound is a constant:  $\mathcal{B}_2^+ = \frac{K-1}{2}$ . In this situation, the proportions of the clusters do not matter. Only getting a Dirac model is sufficient to maximise the OvA GEMINI.

We can further conclude that the bound is tight when considering a Dirac model since  $1^\alpha = 1$  and  $0^\alpha = 0$ , leading to:

$$\begin{aligned}\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ \mathbf{1}_{[\mathbf{x} \in \mathcal{X}_k]}^\alpha \right] &= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ \mathbf{1}_{[\mathbf{x} \in \mathcal{X}_k]} \right], \\ &= \pi_k.\end{aligned}$$

And so do we conclude:

$$\begin{aligned}\mathcal{I}_{D_\alpha}^{\text{ova}}(\mathbf{x}; y) &= (\alpha(\alpha - 1))^{-1} \left[ -1 + \sum_{k=1}^K \pi_k^{1-\alpha} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y = k|\mathbf{x})^\alpha] \right], \\ &= (\alpha(\alpha - 1))^{-1} \left[ -1 + \sum_{k=1}^K \pi_k^{2-\alpha} \right], \\ &= \mathcal{B}_{]1, +\infty[}(\pi_1, \dots, \pi_K)\end{aligned}$$

#### B.4 Case of a $\alpha \in ]0, 1[$

In this case, the front factor  $(\alpha(\alpha - 1))^{-1}$  is negative. Thus, we are interested in minimising the second term and finding the lower bound. We can already infer:

$$\begin{aligned}p(y = k|\mathbf{x}) &\leq p(y = k|\mathbf{x})^\alpha \\ \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y = k|\mathbf{x})] &\leq \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y = k|\mathbf{x})^\alpha] \\ \pi_k &\leq \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y = k|\mathbf{x})^\alpha].\end{aligned}$$

This lower bound is tight for a Dirac model. We can finally compute for the OvA GEMINI that:

$$\begin{aligned}\mathcal{I}_{D_\alpha}^{\text{ova}}(\mathbf{x}; y) &= (\alpha(\alpha - 1))^{-1} \left[ -1 + \sum_{k=1}^K \pi_k^{1-\alpha} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [p(y = k|\mathbf{x})^\alpha] \right], \\ &\leq (\alpha(\alpha - 1))^{-1} \left[ -1 + \sum_{k=1}^K \pi_k^{2-\alpha} \right], \\ &\leq \mathcal{B}_{]0, 1[}.\end{aligned}$$

Hence, we conclude that  $\mathcal{B}_{]0, 1[} = \mathcal{B}_{]1, +\infty[} = \mathcal{B}_{\mathbb{R}^+ \setminus \{1\}}$ . In both cases, using a Dirac model implies that the OvA GEMINI reaches its upper bound.

#### B.5 Case of a negative $\alpha$

The upper bound of the OvA GEMINI in this case is the infinity. Indeed, taking the example of the Dirac model is sufficient to consider regions of the data space  $\mathcal{X}$  where the clustering distribution has no support. Thus, the expectation is undefined, or rather drifts towards infinity.

#### B.6 Specific case of $\alpha = 1$ , the KL divergence

In this case, we need to start the computations all over again using the definition  $f(t) = t \log t$ . Indeed, we can skip the term  $-t + 1$  since it does not affect the value of an  $f$ -divergence, i.e. for any convex function  $f$  s.t.  $f(1) = 0$  and for any real constant  $c$ :

$$D_{f(t)}(p||q) = D_{f(t)+c(t-1)}(p||q).$$

We thus get:

$$\begin{aligned}\mathcal{I}_{D_1}^{\text{ova}}(\mathbf{x}; y) &= \mathbb{E}_{y \sim p(y)} \left[ \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ \frac{p(\mathbf{x}|y = k)}{p(\mathbf{x})} \log \frac{p(\mathbf{x}|y = k)}{p(\mathbf{x})} \right] \right], \\ &= \sum_{k=1}^K \pi_k \int_{\mathcal{X}} p(\mathbf{x}|y = k) \log \left( \frac{p(\mathbf{x}|y = k)}{p(\mathbf{x})} \right) d\mathbf{x}, \\ &= \sum_{k=1}^K \int_{\mathcal{X}} p(y = k|\mathbf{x}) p(\mathbf{x}) \log \left( \frac{p(y = k|\mathbf{x})}{\pi_k} \right) d\mathbf{x}.\end{aligned}$$

We can then separate the log term to make appear the two different entropies contributing to the mutual information:

$$\begin{aligned}\mathcal{I}_{D_1}^{\text{ova}}(\mathbf{x}; y) &= \sum_{k=1}^K \int_{\mathcal{X}} p(y = k|\mathbf{x})p(\mathbf{x}) \log(p(y = k|\mathbf{x}))d\mathbf{x} - \log \pi_k \int_{\mathcal{X}} p(y = k|\mathbf{x})p(\mathbf{x})d\mathbf{x}, \\ &= \sum_{k=1}^K \int_{\mathcal{X}} p(y = k|\mathbf{x})p(\mathbf{x}) \log(p(y = k|\mathbf{x}))d\mathbf{x} - \pi_k \log \pi_k\end{aligned}$$

We find once again an upper bound on the integral depending on  $p(y|\mathbf{x})$ . We know that the function  $g : t \mapsto t \log t$  is convex and below 0 for  $t \in [0, 1]$ . Hence:

$$p(y = k|\mathbf{x}) \log p(y = k|\mathbf{x}) \leq 0,$$

with strict equality iff  $p(y = k|\mathbf{x}) \in \{0, 1\}$ . This implies that the Dirac model maximises the left integral. We deduce the upper bound of the mutual information:

$$\begin{aligned}\mathcal{I}_{D_1}^{\text{ova}}(\mathbf{x}; y) &= \sum_{k=1}^K \int_{\mathcal{X}} p(y = k|\mathbf{x})p(\mathbf{x}) \log(p(y = k|\mathbf{x}))d\mathbf{x} - \pi_k \log \pi_k, \\ &\leq \sum_{k=1}^K -\pi_k \log \pi_k, \\ &\leq \mathcal{B}_1.\end{aligned}$$

This shows that the cluster proportion entropy is the upper bound of the mutual information. It is reached for any Dirac model.

### B.7 Last subcase: the null alpha

In this case, the  $\alpha$ -divergence is defined by the function  $f(t) = -\log t$ . Let us derive again the OvA GEMINI:

$$\begin{aligned}\mathcal{I}_{D_0}^{\text{ova}}(\mathbf{x}; y) &= \mathbb{E}_{y \sim p(y)} \left[ \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ -\log \frac{p(\mathbf{x}|y = k)}{p(\mathbf{x})} \right] \right], \\ &= -\sum_{k=1}^K \pi_k \int_{\mathcal{X}} p(\mathbf{x}) \log \left( \frac{p(\mathbf{x}|y = k)}{p(\mathbf{x})} \right) d\mathbf{x}, \\ &= -\sum_{k=1}^K \pi_k \int_{\mathcal{X}} p(\mathbf{x}) \log \left( \frac{p(y = k|\mathbf{x})}{\pi_k} \right) d\mathbf{x}.\end{aligned}$$

We expand again the logarithm and compute the integral over constant terms factorised by  $p(\mathbf{x})$ :

$$\mathcal{I}_{D_0}^{\text{ova}}(\mathbf{x}; y) = \sum_{k=1}^K \pi_k \log \pi_k - \pi_k \int_{\mathcal{X}} p(\mathbf{x}) \log p(y = k|\mathbf{x})d\mathbf{x}.$$

Now we can see that this OvA GEMINI may converge to infinity. Indeed, for the example of the Dirac model, we evaluate the integral with terms worth  $\lim_{t \rightarrow 0} \log t$ . We cannot conclude on the upper bounds of this case.

### B.8 Maximal upper bound

We have shown so far that for  $\alpha > 0$ , we can derive two different upper bounds that only depend on the proportions of the clusters  $\pi_k$ . These upper bounds can be reached by Dirac model of type  $p(y = k|\mathbf{x}) = \mathbf{1}_{[\mathbf{x} \in \mathcal{X}_k]}$ .

We can now question for the two upper bounds,  $\mathcal{B}_{\mathbb{R}^{+*} \setminus \{1\}}$  and  $\mathcal{B}_1$  what are the optimal cluster proportions  $\pi_k$ . By adding a Lagrangian constraint to enforce  $\sum_{k=1}^K \pi_k = 1$  in each upper bound, we can show that the maximal upper bound is reached iff  $\pi_k = K^{-1} \forall k$ . This concludes our proof.

## APPENDIX C

### PROOF OF PROPOSITION 3.3

Let us consider the value of the OvO GEMINI when the distance  $D$  is an  $f$ -divergence or an IPM.

### C.1 Demonstration for $f$ -divergences

We first need to highlight that  $f$ -divergences come with a conjugate convex function  $g$ . This conjugate enables the inversion of the arguments of the  $f$ -divergence:

$$D_f(p\|q) = D_g(q\|p),$$

for any distributions  $p$  and  $q$ . We can use this trick to revert first the  $f$ -divergence between the distribution  $p(y = k|\mathbf{x})$  and  $p(\mathbf{x})$ :

$$D_f(p(\mathbf{x}|y = k)\|p(\mathbf{x})) = D_g(p(\mathbf{x})\|p(\mathbf{x}|y = k)).$$

We then write  $p(\mathbf{x})$  as a sum marginalising the  $y$  variable. Using the convexity of the function  $g$ , we get a weighted upper bound of this divergence:

$$\begin{aligned} D_g(p(\mathbf{x})\|p(\mathbf{x}|y = k)) &= D_g\left(\sum_{k'=1}^K p(y = k')p(\mathbf{x}|y = k')\| \left(\sum_{k'=1}^K p(y = k')\right)p(\mathbf{x}|y = k)\right), \\ &\leq \sum_{k'=1}^K p(y = k')D_g(p(\mathbf{x}|y = k')\|p(\mathbf{x}|y = k)), \\ &\leq \mathbb{E}_{k' \sim p(y)} [D_g(p(\mathbf{x}|y = k')\|p(\mathbf{x}|y = k))]. \end{aligned}$$

To retrieve the OvO form, we can compute the expectation of this inequality over all possible combinations of  $p(y)$ :

$$\begin{aligned} \mathbb{E}_{y \sim p(y)} [D_f(p(\mathbf{x})\|p(\mathbf{x}|y = k))] &\leq \mathbb{E}_{y_1, y_2 \sim p(y)} [D_g(p(\mathbf{x}|y_1)\|p(\mathbf{x}|y_2))], \\ \mathcal{I}_{D_f}^{\text{OvA}}(\mathbf{x}; y) &\leq \mathcal{I}_{D_g}^{\text{OvO}}(\mathbf{x}; y). \end{aligned}$$

Then, owing to the conjugate convex functions, we can observe that for any  $k, k' \in \{1, \dots, K\}$ :

$$\begin{aligned} D_g(p(\mathbf{x}|y = k)\|p(\mathbf{x}|y = k')) + D_g(p(\mathbf{x}|y = k')\|p(\mathbf{x}|y = k)) &= D_f(p(\mathbf{x}|y = k')\|p(\mathbf{x}|y = k)) \\ &\quad + D_f(p(\mathbf{x}|y = k)\|p(\mathbf{x}|y = k')). \end{aligned}$$

Consequently, the symmetry of OvO in its double expectation implies that:

$$\mathcal{I}_{D_f}^{\text{OvO}}(\mathbf{x}; y) = \mathcal{I}_{D_g}^{\text{OvO}}(\mathbf{x}; y).$$

And so do we conclude that:

$$\mathcal{I}_{D_f}^{\text{OvA}}(\mathbf{x}; y) \leq \mathcal{I}_{D_f}^{\text{OvO}}(\mathbf{x}; y). \quad (23)$$

### C.2 Demonstration for IPMs

For IPMs, we start from the OvA distance between the distribution of an arbitrary cluster  $i$  among  $K$ :

$$D_{\text{IPM}}(p(\mathbf{x}|y = i)\|p(\mathbf{x})) = \sup_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=i)}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{x})].$$

We can extend the expectation of the data distribution over all clusters using the sum rules of probabilities:

$$\begin{aligned} D_{\text{IPM}}(p(\mathbf{x}|y = i)\|p(\mathbf{x})) &= \sup_{f \in \mathcal{F}} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=i)}[f(\mathbf{x})] - \sum_{k=1}^K p(y = k) \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=k)}[f(\mathbf{x})], \\ &= \sup_{f \in \mathcal{F}} \sum_{k=1}^K p(y = k) (\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=i)}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=k)}[f(\mathbf{x})]). \end{aligned}$$

We used in the second line the fact that the sum of  $\sum_{k=1}^K p(y = k) = 1$  to factorise the first expectation. Finally, we can use the property that the supremum of a sum is lower or equal than a sum of suprema:

$$\begin{aligned} D_{\text{IPM}}(p(\mathbf{x}|y = i)\|p(\mathbf{x})) &\leq \sum_{k=1}^K \sup_{f \in \mathcal{F}} p(y = k) (\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=i)}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=k)}[f(\mathbf{x})]), \\ &\leq \sum_{k=1}^K p(y = k) D_{\text{IPM}}(p(\mathbf{x}|y = i)\|p(\mathbf{x}|y = k)). \end{aligned}$$

This upper bound corresponds to the expectation of the IPM between a specific cluster distribution  $i$  and all other cluster distributions. Finally, by performing the expectation over all cluster of index  $i$ , we can conclude that:

$$\mathcal{I}_{\text{IPM}}^{\text{ova}}(\mathbf{x}; y) \leq \mathcal{I}_{\text{IPM}}^{\text{ovo}}(\mathbf{x}; y). \quad (24)$$

## APPENDIX D PROOF OF PROPOSITION 3.4

We will show here that when the cluster variable  $y$  is binary, the OvA and OvO GEMINIs are equal if we use IPMs for distance between distributions. Indeed we can first unfold both equations:

$$\begin{aligned} \mathcal{I}_{\text{IPM}}^{\text{ova}}(\mathbf{x}, y) &= \mathbb{E}_{y \sim p(y)} [D_{\text{IPM}}(p(\mathbf{x}|y) \| p(\mathbf{x}))] \\ &= p(y=0)D_{\text{IPM}}(p(\mathbf{x}|y=0) \| p(\mathbf{x})) + p(y=1)D_{\text{IPM}}(p(\mathbf{x}|y=1) \| p(\mathbf{x})) \\ &= p(y=0) \sup_{f \in \mathcal{F}} \{ \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=0)} [f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [f(\mathbf{x})] \} \\ &\quad + p(y=1) \sup_{g \in \mathcal{F}} \{ \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=1)} [g(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [g(\mathbf{x})] \}, \end{aligned} \quad (25)$$

and for the OvO, we simply use the symmetric property of IPMs:

$$\begin{aligned} \mathcal{I}_{\text{IPM}}^{\text{ovo}}(\mathbf{x}, y) &= \mathbb{E}_{y_a, y_b \sim p(y)} [D_{\text{IPM}}(p(\mathbf{x}|y_a) \| p(\mathbf{x}|y_b))] \\ &= p(y=0)p(y=1)D_{\text{IPM}}(p(\mathbf{x}|y=0) \| p(\mathbf{x}|y=1)) \\ &\quad + p(y=1)p(y=0)D_{\text{IPM}}(p(\mathbf{x}|y=1) \| p(\mathbf{x}|y=0)) \\ &= 2p(y=0)p(y=1)D_{\text{IPM}}(p(\mathbf{x}|y=0) \| p(\mathbf{x}|y=1)). \end{aligned} \quad (26)$$

Notice that in Eq. (26), we skipped the terms where both the random variables  $y_1$  and  $y_2$  are equal, since the implied distance is necessarily 0.

Now, to show the equivalence of both equations (25) and (26), we simply need to write the sum rule of probabilities leading to the marginalisation of  $\mathbf{x}$ :

$$p(\mathbf{x}) = p(\mathbf{x}|y=0)p(y=0) + p(\mathbf{x}|y=1)p(y=1).$$

Thus, we can rewrite the expectations depending on the distribution  $p(\mathbf{x})$  with other distributions for any function  $f$ :

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [f(\mathbf{x})] = p(y=0)\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=0)} [f(\mathbf{x})] + p(y=1)\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=1)} [f(\mathbf{x})],$$

which we can incorporate back into Eq. (25) to get:

$$\begin{aligned} \mathcal{I}_{\text{IPM}}^{\text{ova}}(\mathbf{x}, y) &= p(y=0) \sup_{f \in \mathcal{F}} \{ (1 - p(y=0))\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=0)} [f(\mathbf{x})] - p(y=1)\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=1)} [f(\mathbf{x})] \} \\ &\quad + p(y=1) \sup_{g \in \mathcal{F}} \{ (1 - p(y=1))\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=1)} [g(\mathbf{x})] - p(y=0)\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=0)} [g(\mathbf{x})] \}. \end{aligned}$$

Since we only use two clusters, we know that  $p(y=1) = 1 - p(y=0)$ . This helps us factorising terms inside the sup expressions:

$$\begin{aligned} \mathcal{I}_{\text{IPM}}^{\text{ova}}(\mathbf{x}, y) &= p(y=0) \sup_{f \in \mathcal{F}} \{ p(y=1) [\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=0)} [f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=1)} [f(\mathbf{x})]] \} \\ &\quad + p(y=1) \sup_{g \in \mathcal{F}} \{ p(y=0) [\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=1)} [g(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=0)} [g(\mathbf{x})]] \}. \end{aligned}$$

Eventually, the factors  $p(y=0)$  and  $p(y=1)$  do not depend on the functions  $f$  and  $g$ , so we can pull them out of the supremum. The remaining expressions are then symmetric and can be thus factorised:

$$\begin{aligned} \mathcal{I}_{\text{IPM}}^{\text{ova}}(\mathbf{x}, y) &= 2p(y=0)p(y=1) \sup_{f \in \mathcal{F}} \{ \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=0)} [f(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}|y=1)} [f(\mathbf{x})] \} \\ &= 2p(y=0)p(y=1)D_{\text{IPM}}(p(\mathbf{x}|y=0) \| p(\mathbf{x}|y=1)) \\ &= \mathcal{I}_{\text{IPM}}^{\text{ovo}}(\mathbf{x}, y) \end{aligned}$$

This concludes the proof.

## APPENDIX E

### PROOF OF PROPOSITION 3.5

For any  $f$ -divergence and two distribution  $p$  and  $q$  taking value in the space  $\mathcal{X}$ , then disjoint support between  $p$  and  $q$  implies the maximisation of the  $f$ -divergence. Indeed, the bounds of an  $f$ -divergence are:

$$0 \leq D_f(p, q) \leq f(0) + g(0),$$

where the upper bound can be infinity depending on  $f$  and its convex conjugate  $g : t \rightarrow tf(1/t)$ . Thus, for any two different clusters  $k \neq k'$ :

$$0 \leq D_f(p(\mathbf{x}|y = k)||p(\mathbf{x}|y = k')) \leq f(0) + g(0).$$

However, distributions for the same clusters have an  $f$ -divergence of 0. We can therefore sum all the terms and their respective upper bounds:

$$\begin{aligned} \sum_{k \neq k'}^K p(y = k)p(y = k')D_f(p(\mathbf{x}|y = k)||p(\mathbf{x}|y = k')) &\leq \sum_{k \neq k'}^K p(y = k)p(y = k')(f(0) + g(0)) \\ \mathcal{I}_{D_f}^{\text{ovO}}(\mathbf{x}, y) &\leq \sum_{k=1}^K p(y = k)p(y \neq k)(f(0) + g(0)), \end{aligned}$$

Following [9, theorem 5], disjoint supports between the distribution  $p(\mathbf{x}|y = k)$  and  $p(\mathbf{x}|y = k')$  implies the equality with the upper bound. Assume that the data space  $\mathcal{X}$  is separated into  $K$  disjoint and supplementary spaces  $\mathcal{X}_k$ . To each subspace  $\mathcal{X}_k$  corresponds a cluster distribution  $p(\mathbf{x}|y = k)$ . This disjoint supports are achieved for any model of the form:

$$p(y = k|\mathbf{x}) = \mathbf{1}[\mathbf{x} \in \mathcal{X}_k],$$

which implies the disjoint distributions:

$$p(\mathbf{x}|y = k) \propto \mathbf{1}[\mathbf{x} \in \mathcal{X}_k]$$

Each of these spaces control the proportion of data in the cluster  $k$ , and hence controls  $p(y = k)$ . Thus, the OvO GEMINI is equal to its upper bound owing to disjoint supports:

$$\mathcal{I}_{D_f}^{\text{ovO}}(\mathbf{x}, y) = \sum_{k=1}^K p(y = k)p(y \neq k)(f(0) + g(0))$$

We need to maximise the upper bound. This will be the maximum value of OvO GEMINI reachable for models with disjoint supports. Adding a Lagrangian term to respect the constraint of  $\sum_{k=1}^K p(y = k) = 1$  leads to the optimal solution  $p(y = k) = \frac{1}{K}$ . This concludes the proof.

## APPENDIX F

### DERIVING GEMINIS

We show in this appendix how to derive all estimable forms of the GEMINI.

#### F.1 $f$ -divergence GEMINI

We detail here the derivation for 3  $f$ -divergences that we previously chose: the KL divergence, the TV distance and the squared Hellinger distance, as well as the generic scenario for any function  $f$ .

##### F.1.1 Generic scenario

First, we recall that the definition of an  $f$ -divergence involves a convex function:

$$\begin{aligned} f : \mathbb{R}^+ &\rightarrow \mathbb{R} \\ x &\rightarrow f(x) \\ \text{s.t. } f(1) &= 0, \end{aligned}$$

between two distributions  $p$  and  $q$  as described:

$$D_{f\text{-div}}(p, q) = \mathbb{E}_{\mathbf{x} \sim q} \left[ f \left( \frac{p(\mathbf{x})}{q(\mathbf{x})} \right) \right].$$



We simply inject this definition in the OvA GEMINI and directly obtain both an expectation on the cluster assignment  $y$  and on the data variable  $\mathbf{x}$ . We then merge the writing of the two expectations for the sake of clarity.

$$\begin{aligned}\mathcal{I}_{f\text{-div}}^{\text{ova}}(x; y) &= \mathbb{E}_{p_\theta(y)} [D_{f\text{-div}}(p_\theta(\mathbf{x}|y) || p_{\text{data}}(\mathbf{x}))], \\ &= \mathbb{E}_{p_\theta(y)} \left[ \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[ f \left( \frac{p_\theta(\mathbf{x}|y)}{p_{\text{data}}(\mathbf{x})} \right) \right] \right], \\ &= \mathbb{E}_{p_\theta(y), p_{\text{data}}(\mathbf{x})} \left[ f \left( \frac{p_\theta(y|\mathbf{x})}{p_\theta(y)} \right) \right].\end{aligned}$$

Injecting the  $f$ -divergence in the OvO GEMINI first yields:

$$\begin{aligned}\mathcal{I}_{f\text{-div}}^{\text{ovo}}(x; y) &= \mathbb{E}_{p_\theta(y_a), p_\theta(y_b)} [D_{f\text{-div}}(p_\theta(\mathbf{x}|y_a) || p_\theta(\mathbf{x}|y_b))], \\ &= \mathbb{E}_{p_\theta(y_a), p_\theta(y_b)} \left[ \mathbb{E}_{p_\theta(\mathbf{x}|y_b)} \left[ f \left( \frac{p_\theta(\mathbf{x}|y_a)}{p_\theta(\mathbf{x}|y_b)} \right) \right] \right].\end{aligned}$$

Now, by using Bayes theorem, we can perform the inner expectation over the data distribution. We then merge the expectations for the sake of clarity.

$$\begin{aligned}\mathcal{I}_{f\text{-div}}^{\text{ovo}}(x; y) &= \mathbb{E}_{p_\theta(y_a), p_\theta(y_b)} \left[ \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} f \left( \frac{p_\theta(\mathbf{x}|y_a)}{p_\theta(\mathbf{x}|y_b)} \right) \right] \right], \\ &= \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} f \left( \frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})p_\theta(y_a)} \right) \right].\end{aligned}$$

Notice that we also changed the ratio of conditional distributions inside the function by a ratio of posteriors through Bayes' theorem, weighted by the relative cluster proportions.

Now, we can derive into details these equations for the 3  $f$ -divergences we focused on: the KL divergence, the TV distance and the squared Hellinger distance.

### F.1.2 Kullback-Leibler divergence

The function for Kullback-Leibler is  $f(t) = t \log t$ . We do not need to write the OvA equation: it is straightforwardly the usual MI. For the OvO, we inject the function definition by replacing:

$$t = \frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})p_\theta(y_a)},$$

in order to get:

$$\mathcal{I}_{\text{KL}}^{\text{ovo}}(x; y) = \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} \times \frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})p_\theta(y_a)} \log \frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})p_\theta(y_a)} \right].$$

We can first simplify the factors outside of the logs:

$$\mathcal{I}_{\text{KL}}^{\text{ovo}}(x; y) = \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} \log \frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})p_\theta(y_a)} \right].$$

If we use the properties of the log, we can separate the inner term in two sub-expressions:

$$\mathcal{I}_{\text{KL}}^{\text{ovo}}(x; y) = \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} \log \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} + \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} \log \frac{p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})} \right].$$

Hence, we can use the linearity of the expectation to separate the two terms above. The first term is constant w.r.t.  $y_b$ , so we can remove this variable from the expectation among the subscripts:

$$\mathcal{I}_{\text{KL}}^{\text{ovo}}(\mathbf{x}, y) = \mathbb{E}_{p_\theta(y_a), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} \log \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} \right] + \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} \log \frac{p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})} \right].$$

Since the variables  $y_a$  and  $y_b$  are independent, we can use the fact that:

$$\mathbb{E}_{p_\theta(y_a)} \left[ \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} \right] = \int p_\theta(y_a) \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} dy_a = 1,$$

inside the second term to reveal the final form of the equation:

$$\mathcal{I}_{\text{KL}}^{\text{ovo}}(x; y) = \mathbb{E}_{p_{\text{data}}(\mathbf{x}), p_\theta(y)} \left[ \frac{p_\theta(y|\mathbf{x})}{p_\theta(y)} \log \frac{p_\theta(y|\mathbf{x})}{p_\theta(y)} \right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x}), p_\theta(y)} \left[ \log \frac{p_\theta(y)}{p_\theta(y|\mathbf{x})} \right].$$

Notice that since both terms did not compare one cluster assignment  $y_a$  against another  $y_b$ , we can switch to the same common variable  $y$ . Both terms are in fact KL divergences depending on the cluster assignment  $y$ . The first is the reverse of the second. This sum of KL divergences is sometimes called the *symmetric KL*, and so can we write in two ways the OvO KL-GEMINI:

$$\begin{aligned} \mathcal{I}_{\text{KL}}^{\text{ovo}}(x; y) &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [D_{\text{KL}}(p_\theta(y|\mathbf{x})||p_\theta(y))] + \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [D_{\text{KL}}(p_\theta(y)||p_\theta(y|\mathbf{x}))], \\ &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} [D_{\text{KL-sym}}(p_\theta(y|\mathbf{x})||p_\theta(y))]. \end{aligned}$$

We can also think of this equation as the usual MI with an additional term based on the reversed KL divergence.

### F.1.3 Total Variation distance

For the total variation, the function is  $f(t) = \frac{1}{2}|t - 1|$ . Thus, the OvA GEMINI is:

$$\mathcal{I}_{\text{TV}}^{\text{ova}}(x; y) = \frac{1}{2} \mathbb{E}_{p_\theta(y), p_{\text{data}}(\mathbf{x})} \left[ \left| \frac{p_\theta(y|\mathbf{x})}{p_\theta(y)} - 1 \right| \right].$$

And the OvO is:

$$\begin{aligned} \mathcal{I}_{\text{TV}}^{\text{ovo}}(x; y) &= \frac{1}{2} \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} \left| \frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})p_\theta(y_a)} - 1 \right| \right], \\ &= \frac{1}{2} \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \left| \frac{p_\theta(y_a|\mathbf{x})}{p_\theta(y_a)} - \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} \right| \right]. \end{aligned}$$

We did not find any further simplification of these equations.

### F.1.4 Squared Hellinger distance

Finally, the squared Hellinger distance is based on  $f(t) = 2(1 - \sqrt{t})$ . Hence the OvA unfolds as:

$$\begin{aligned} \mathcal{I}_{\text{H}^2}^{\text{ova}}(x; y) &= \mathbb{E}_{p_\theta(y), p_{\text{data}}(\mathbf{x})} \left[ 2 \left( 1 - \sqrt{\frac{p_\theta(y|\mathbf{x})}{p_\theta(y)}} \right) \right], \\ &= 2 - 2 \mathbb{E}_{p_{\text{data}}(\mathbf{x}), p_\theta(y)} \left[ \sqrt{\frac{p_\theta(y|\mathbf{x})}{p_\theta(y)}} \right]. \end{aligned}$$

The idea of the squared OvA Hellinger-GEMINI is therefore to minimise the expected square root of the relative certainty between the posterior and cluster proportion.

For the OvO setting, the definition yields:

$$\mathcal{I}_{\text{H}^2}^{\text{ovo}}(x; y) = \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} \times 2 \times \left( 1 - \sqrt{\frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b)}{p_\theta(y_b|\mathbf{x})p_\theta(y_a)}} \right) \right],$$

which we can already simplify by putting the constant 2 outside of the expectation, and by inserting all factors inside the square root before simplifying and separating the expectation:

$$\begin{aligned} \mathcal{I}_{\text{H}^2}^{\text{ovo}}(x; y) &= 2 \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} - \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} \sqrt{\frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b)}{p_\theta(y_a)p_\theta(y_b|\mathbf{x})}} \right], \\ &= 2 \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_b|\mathbf{x})}{p_\theta(y_b)} \right] - 2 \mathbb{E}_{p_\theta(y_a), p_\theta(y_b), p_{\text{data}}(\mathbf{x})} \left[ \sqrt{\frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b|\mathbf{x})}{p_\theta(y_a)p_\theta(y_b)}} \right]. \end{aligned}$$

We can replace the first term by the constant 1, as shown for the OvO KL derivation. Since we can split the square root into the product of two square roots, we can apply twice the expectation over  $y_a$  and  $y_b$  because these variables are independent:

$$\mathcal{I}_{H^2}^{\text{OvO}}(x; y) = 2 - 2\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[ \mathbb{E}_{p_{\theta}(y)} \left[ \sqrt{\frac{p_{\theta}(y|\mathbf{x})}{p_{\theta}(y)}}} \right]^2 \right].$$

To avoid computing this squared expectation, we use the equation of the variance  $\mathbb{V}$  to replace it. Thus:

$$\begin{aligned} \mathcal{I}_{H^2}^{\text{OvO}}(x; y) &= 2 - 2\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[ \mathbb{E}_{p_{\theta}(y)} \left[ \frac{p_{\theta}(y|\mathbf{x})}{p_{\theta}(y)} \right] - \mathbb{V}_{p_{\theta}(y)} \left[ \sqrt{\frac{p_{\theta}(y|\mathbf{x})}{p_{\theta}(y)}}} \right] \right], \\ &= 2 - 2\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[ \mathbb{E}_{p_{\theta}(y)} \left[ \frac{p_{\theta}(y|\mathbf{x})}{p_{\theta}(y)} \right] \right] + 2\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[ \mathbb{V}_{p_{\theta}(y)} \left[ \sqrt{\frac{p_{\theta}(y|\mathbf{x})}{p_{\theta}(y)}}} \right] \right]. \end{aligned}$$

Then, for the same reason as before, the second term is worth 1, which cancels the first constant. We therefore end up with:

$$\mathcal{I}_{H^2}^{\text{OvO}}(x; y) = 2\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[ \mathbb{V}_{p_{\theta}(y)} \left[ \sqrt{\frac{p_{\theta}(y|\mathbf{x})}{p_{\theta}(y)}}} \right] \right].$$

Similar to the OvO KL case, the OvO squared Hellinger converges to an OvA setting, i.e. we only need information about the cluster distribution itself without comparing it to another. Furthermore, the idea of maximising the variance of the cluster assignments is straightforward for clustering.

## F.2 Maximum Mean Discrepancy

When using an IPM with a family of functions that project an input of  $\mathcal{X}$  to the unit ball of an RKHS  $\mathcal{H}$ , the IPM becomes the MMD distance.

$$\begin{aligned} \text{MMD}(p, q) &= \sup_{f: \|f\|_{\mathcal{H}} \leq 1} \mathbb{E}_{\mathbf{x}_a \sim p} [f(\mathbf{x}_a)] - \mathbb{E}_{\mathbf{x}_b \sim q} [f(\mathbf{x}_b)], \\ &= \|\mathbb{E}_{\mathbf{x}_a \sim p} [\varphi(\mathbf{x}_a)] - \mathbb{E}_{\mathbf{x}_b \sim q} [\varphi(\mathbf{x}_b)]\|_{\mathcal{H}}, \end{aligned}$$

where  $\varphi$  is an embedding function of the RKHS.

By using a kernel function  $\kappa(\mathbf{x}_a, \mathbf{x}_b) = \langle \varphi(\mathbf{x}_a), \varphi(\mathbf{x}_b) \rangle$ , we can express the square of this distance thanks to inner product space properties [27]:

$$\text{MMD}^2(p, q) = \mathbb{E}_{\mathbf{x}_a, \mathbf{x}'_a \sim p} [\kappa(\mathbf{x}_a, \mathbf{x}'_a)] + \mathbb{E}_{\mathbf{x}_b, \mathbf{x}'_b \sim q} [\kappa(\mathbf{x}_b, \mathbf{x}'_b)] - 2\mathbb{E}_{\mathbf{x}_a \sim p, \mathbf{x}_b \sim q} [\kappa(\mathbf{x}_a, \mathbf{x}_b)].$$

Now, we can derive each term of this equation using our distributions  $p \equiv p_{\theta}(\mathbf{x}|y)$  and  $q \equiv p_{\text{data}}(\mathbf{x})$  for the OvA case, and  $p \equiv p_{\theta}(\mathbf{x}|y_a)$ ,  $q \equiv p_{\theta}(\mathbf{x}|y_b)$  for the OvO case. In both scenarios, we aim at finding an expectation over the data variable  $x$  using only the respectively known and estimable terms  $p_{\theta}(y|\mathbf{x})$  and  $p_{\theta}(y)$ .

F.2.0.1 OvA scenario: For the first term, we use Bayes' theorem twice to get an expectation over two variables  $\mathbf{x}_a$  and  $\mathbf{x}_b$  drawn from the data distribution.

$$\begin{aligned} \mathbb{E}_{\mathbf{x}_a, \mathbf{x}'_a \sim p} &= \mathbb{E}_{\mathbf{x}_a, \mathbf{x}'_a \sim p_{\theta}(\mathbf{x}|y)} [\kappa(\mathbf{x}_a, \mathbf{x}'_a)], \\ &= \mathbb{E}_{\mathbf{x}_a, \mathbf{x}'_a \sim p_{\text{data}}(\mathbf{x})} \left[ \frac{p_{\theta}(y|\mathbf{x}_a)p_{\theta}(y|\mathbf{x}'_a)}{p_{\theta}(y)^2} \kappa(\mathbf{x}_a, \mathbf{x}'_a) \right]. \end{aligned}$$

For the second term, we do not need to perform anything particular as we directly get an expectation over the data variables  $\mathbf{x}_a$  and  $\mathbf{x}_b$ .

$$\mathbb{E}_{\mathbf{x}_b, \mathbf{x}'_b \sim q} = \mathbb{E}_{\mathbf{x}_b, \mathbf{x}'_b \sim p_{\text{data}}(\mathbf{x})} [\kappa(\mathbf{x}_b, \mathbf{x}'_b)].$$

The last term only needs Bayes theorem once, for the distribution  $q$  is directly replaced by the data distribution  $p_{\text{data}}(\mathbf{x})$ :

$$\begin{aligned} \mathbb{E}_{\mathbf{x}_a \sim p, \mathbf{x}_b \sim q} &= \mathbb{E}_{\mathbf{x}_a \sim p_{\theta}(\mathbf{x}|y), \mathbf{x}_b \sim p_{\text{data}}(\mathbf{x})} [\kappa(\mathbf{x}_a, \mathbf{x}_b)], \\ &= \mathbb{E}_{\mathbf{x}_a, \mathbf{x}_b \sim p_{\text{data}}(\mathbf{x})} \left[ \frac{p_{\theta}(y|\mathbf{x}_a)}{p_{\theta}(y)} \kappa(\mathbf{x}_a, \mathbf{x}_b) \right]. \end{aligned}$$

Note that for the last term, we could replace  $p_\theta(y|\mathbf{x}_a)$  by  $p_\theta(y|\mathbf{x}_b)$ ; that would not affect the result since  $\mathbf{x}_a$  and  $\mathbf{x}_b$  are independently drawn from  $p_{\text{data}}(\mathbf{x})$ . We thus replace all terms, and do not forget to put a square root on the entire sum since we have computed so far the squared MMD:

$$\begin{aligned}\mathcal{I}_{\text{MMD}}^{\text{ova}}(x; y) &= \mathbb{E}_{p_\theta(y)} [\text{MMD}(p_\theta(\mathbf{x}|y), p_{\text{data}}(\mathbf{x}))], \\ &= \mathbb{E}_{p_\theta(y)} \left[ \left( \mathbb{E}_{\mathbf{x}_a, \mathbf{x}'_a \sim p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y|\mathbf{x}_a)p_\theta(y|\mathbf{x}'_a)}{p_\theta(y)^2} \kappa(\mathbf{x}_a, \mathbf{x}'_a) \right] \right. \right. \\ &\quad \left. \left. + \mathbb{E}_{\mathbf{x}_b, \mathbf{x}'_b \sim p_{\text{data}}(\mathbf{x})} [\kappa(\mathbf{x}_b, \mathbf{x}'_b)] - 2\mathbb{E}_{\mathbf{x}_a, \mathbf{x}_b \sim p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y|\mathbf{x}_a)}{p_\theta(y)} \kappa(\mathbf{x}_a, \mathbf{x}_b) \right] \right) \right]^{\frac{1}{2}}.\end{aligned}$$

Since all variables  $\mathbf{x}_a$ ,  $\mathbf{x}'_a$ ,  $\mathbf{x}_b$  and  $\mathbf{x}'_b$  are independently drawn from the same distribution  $p_{\text{data}}(\mathbf{x})$ , we can replace all of them by the variables  $\mathbf{x}$  and  $\mathbf{x}'$ . We then use the linearity of the expectation and factorise by the kernel  $\kappa(\mathbf{x}, \mathbf{x}')$ :

$$\mathcal{I}_{\text{MMD}}^{\text{ova}}(x; y) = \mathbb{E}_{p_\theta(y)} \left[ \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim p_{\text{data}}(\mathbf{x})} \left[ \kappa(\mathbf{x}, \mathbf{x}') \left( \frac{p_\theta(y|\mathbf{x})p_\theta(y|\mathbf{x}')}{p_\theta(y)^2} + 1 - 2\frac{p_\theta(y|\mathbf{x})}{p_\theta(y)} \right) \right]^{\frac{1}{2}} \right].$$

F.2.0.2 OvO scenario: The two first terms of the OvO MMD are the same as the first term of the OvA setting, with a simple subscript  $a$  or  $b$  at the appropriate place. Only the negative term changes. We once again use Bayes' theorem twice:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}_a \sim p, \mathbf{x}_b \sim q} [\kappa(\mathbf{x}_a, \mathbf{x}_b)] &= \mathbb{E}_{\mathbf{x}_a \sim p_\theta(\mathbf{x}|y_a), \mathbf{x}_b \sim p_\theta(\mathbf{x}|y_b)} [\kappa(\mathbf{x}_a, \mathbf{x}_b)], \\ &= \mathbb{E}_{\mathbf{x}_a, \mathbf{x}_b \sim p_{\text{data}}(\mathbf{x})} \left[ \frac{p_\theta(y_a|\mathbf{x}_a)}{p_\theta(y_a)} \frac{p_\theta(y_b|\mathbf{x}_b)}{p_\theta(y_b)} \kappa(\mathbf{x}_a, \mathbf{x}_b) \right].\end{aligned}$$

The final sum is hence similar to the OvA:

$$\begin{aligned}\mathcal{I}_{\text{MMD}}^{\text{ovo}}(x; y) &= \mathbb{E}_{p_\theta(y_a), p_\theta(y_b)} [\text{MMD}(p_\theta(\mathbf{x}|y_a), p_\theta(\mathbf{x}|y_b))], \\ &= \mathbb{E}_{p_\theta(y_a), p_\theta(y_b)} \left[ \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim p_{\text{data}}(\mathbf{x})} \left[ \kappa(\mathbf{x}, \mathbf{x}') \left( \frac{p_\theta(y_a|\mathbf{x})p_\theta(y_a|\mathbf{x}')}{p_\theta(y_a)^2} + \frac{p_\theta(y_b|\mathbf{x})p_\theta(y_b|\mathbf{x}')}{p_\theta(y_b)^2} \right. \right. \right. \\ &\quad \left. \left. - 2\frac{p_\theta(y_a|\mathbf{x})p_\theta(y_b|\mathbf{x}')}{p_\theta(y_a)p_\theta(y_b)} \right) \right]^{\frac{1}{2}} \right].\end{aligned}$$

### F.3 Wasserstein distance (Proof of Prop. 3.2)

To compute the Wasserstein distance between the distributions  $p_\theta(\mathbf{x}|y = k)$ , we estimate it using approximate distributions. We replace  $p_\theta(\mathbf{x}|y = k)$  by a weighted sum of Dirac measures on specific samples  $\mathbf{x}_i$ :  $p_N^k$ :

$$p_\theta(\mathbf{x}|y = k) \approx \sum_{i=1}^N m_i^k \delta_{\mathbf{x}_i} = p_N^k,$$

where  $\{m_i^k\}_{i=1}^N$  is the set of weights. We now show that computing the Wasserstein distance between these approximates converges to the correct distance. We first need to show that  $p_N^k$  weakly converges to  $p$ . To that end, we will use the Portmanteau theorem [5]. Let  $f$  be any bounded and continuous function. Computing the expectation of such through  $p_\theta$  is:

$$\mathbb{E}_{\mathbf{x} \sim p_\theta(\mathbf{x}|y=k)} [f(\mathbf{x})] = \int_{\mathcal{X}} f(\mathbf{x}) p_\theta(\mathbf{x}|y = k) d\mathbf{x},$$

which can be estimated using self-normalised importance sampling [52, Chapter 9]. The proposal distribution we take for sampling is  $p_{\text{data}}(\mathbf{x})$ . Although we cannot evaluate both  $p_\theta(\mathbf{x}|y)$  and  $p_{\text{data}}(\mathbf{x})$  up to a constant, we can evaluate their ratio up to a constant which is sufficient:

$$\begin{aligned}\mathbb{E}_{\mathbf{x} \sim p_\theta(\mathbf{x}|y=k)} [f(\mathbf{x})] &= \int_{\mathcal{X}} f(\mathbf{x}) \frac{p_\theta(\mathbf{x}|y = k)}{p_{\text{data}}(\mathbf{x})} p_{\text{data}}(\mathbf{x}) d\mathbf{x}, \\ &= \int_{\mathcal{X}} f(\mathbf{x}) \frac{p_\theta(y = k|\mathbf{x})}{p_\theta(y = k)} p_{\text{data}}(\mathbf{x}) d\mathbf{x}, \\ &\approx \sum_{i=1}^N f(\mathbf{x}_i) \frac{p_\theta(y = k|\mathbf{x} = \mathbf{x}_i)}{\sum_{j=1}^N p_\theta(y = k|\mathbf{x} = \mathbf{x}_j)}.\end{aligned}$$

Now, by noticing in the last line that the importance weights are self normalised and add up to 1, we can identify them as the point masses of our previous Dirac approximations:

$$m_i^k = \frac{p_\theta(y = k | \mathbf{x} = \mathbf{x}_i)}{\sum_{j=1}^N p_\theta(y = k | \mathbf{x} = \mathbf{x}_j)}.$$

This allows to write that the Monte Carlo estimation through importance sampling of the expectation w.r.t  $p_\theta(\mathbf{x} | y = k)$  is directly the expectation taken on the discrete approximation  $p_N^k$ . We can conclude that there is a convergence between the two expectations owing to the law of large numbers:

$$\lim_{N \rightarrow +\infty} \mathbb{E}_{\mathbf{x} \sim p_N^k} [f(\mathbf{x})] = \mathbb{E}_{\mathbf{x} \sim p_\theta(\mathbf{x} | y = k)} [f(\mathbf{x})].$$

Since  $f$  is bounded and continuous, the portmanteau theorem [5] states that  $p_N^k$  weakly converges to  $p_\theta(\mathbf{x} | y = k)$  when defining the importance weights as the normalised predictions cluster-wise.

To conclude, when two series of measures  $p_N$  and  $q_N$  weakly converge respectively to  $p$  and  $q$ , so does their Wasserstein distance [65, Corollary 6.9], hence:

$$\lim_{N \rightarrow +\infty} \mathcal{W}_c(p_N^{k_1}, p_N^{k_2}) = \mathcal{W}_c(p_\theta(\mathbf{x} | y = k_1) || p_\theta(\mathbf{x} | y = k_2)). \quad (27)$$

For the one-vs-all Wasserstein GEMINI, we simply need to set the second distribution to the empirical data distribution:  $m_i = 1/N$ .

## APPENDIX G

### OTHER MAXIMISATIONS OF THE WASSERSTEIN DISTANCES

The Wasserstein-1 metric can be considered as an IPM defined over a set of 1-Lipschitz functions. Indeed, such writing is the dual representation of the Wasserstein-1 metric:

$$\mathcal{W}_c(p || q) = \sup_{f, \|f\|_L \leq 1} \mathbb{E}_{x \sim p} [f(x)] - \mathbb{E}_{z \sim q} [f(z)].$$

Yet, evaluating a supremum as an objective to maximise is hardly compatible with the usual backpropagation in neural networks. This definition was used in attempts to stabilise GAN training [1] by using 1-Lipschitz neural networks [25]. However, the Lipschitz continuity was achieved at the time by weight clipping, whereas other methods such as spectral normalisation [48] now allow arbitrarily large weights. The restriction of 1-Lipschitz functions to 1-Lipschitz neural networks does not equal the true Wasserstein distance, and the term "neural net distance" is sometimes preferred [2]. Still, estimating the Wasserstein distance using a set of Lipschitz functions derived from neural networks architectures brings more difficulties to actually leverage the true distance according to the energy cost  $c$  of Eq. 13.

Globally, we hardly experimented the generic IPM for GEMINIs. Our efforts for defining a set of 1-Lipschitz critics, one per cluster for OvA or one per pair of clusters for OvO, to perform the neural net distance [2] were not fruitful. This is mainly because such an objective requires the definition of one neural network for the posterior distribution  $p_\theta(y | \mathbf{x})$  and  $K$  (resp.  $K(K-1)/2$ ) other 1-Lipschitz neural networks for the OvA (resp. OvO) critics, i.e. a large number of parameters. Moreover, this brings the problem of designing not only one but many neural networks while the design of one accurate architecture is already a sufficient problem.

## APPENDIX H

### CHOOSING A GEMINI

The complexity of GEMINI increases with the distances previously mentioned depending on the number of clusters  $K$  and the number of samples per batch  $N$ . It ranges from  $\mathcal{O}(NK)$  for the usual MI to  $\mathcal{O}(K^2 N^3 \log N)$  for the OvO Wasserstein-GEMINI. As an example, we show in Figure 12 the average time of GEMINI as the number of tasked clusters increases for both 10 samples per batch (Figure 12a) and 500 samples (Figure 12b). The batches consist in randomly generated prediction and distances or kernel between randomly generated data.

The OvO Wasserstein is the most complex, and so its usage should remain for 10 clusters or less overall. The second most time-consuming loss is the OvA Wasserstein, however its tendency in optimisation to only find 2 clusters makes it inappropriate. The main difference also to notice between the following MMD is regarding their memory complexity. The OvA MMD requires only  $\mathcal{O}(KN^2)$  while the OvO MMD requires  $\mathcal{O}(K^2 N^2)$ . This memory complexity should be the major guide to choosing one MMD-GEMINI or the other. Thus, the minimal time-consuming and resource-demanding GEMINI is the OvA MMD if we consider GEMINIs that incorporate knowledge of data through kernels and distances. Other versions involving  $f$ -divergences have in fact the same complexity as MI in our implementations, apart from the OvO TV which reaches  $\mathcal{O}(K^2 N)$  in our implementation.

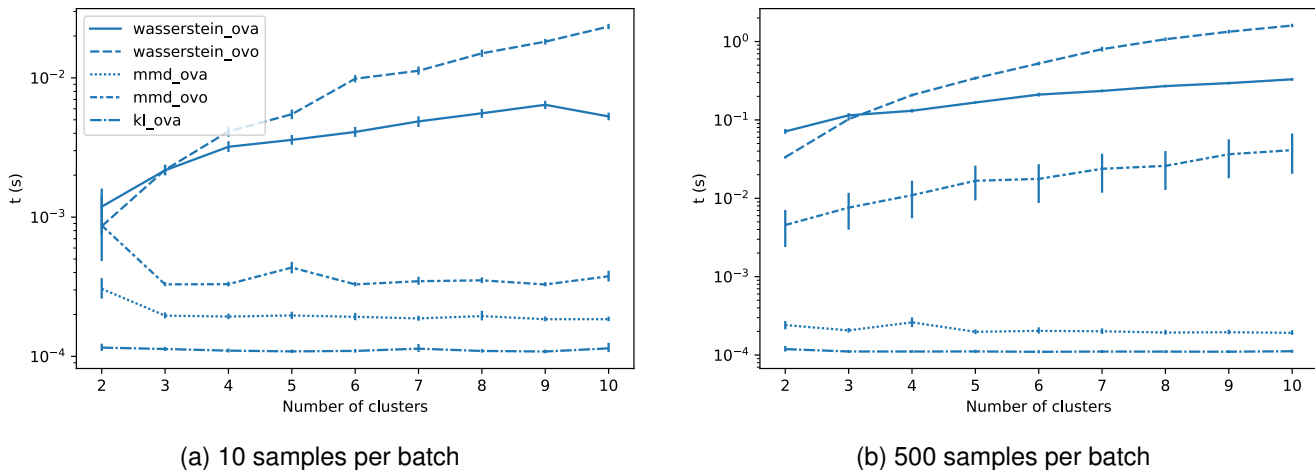


Fig. 12. Average performance time (in seconds) of GEMINIs as the number of tasked clusters grows for batches of size 10 and 500 samples.

## APPENDIX I FURTHER SPEED-UPS FOR THE WASSERSTEIN-OVO

The complexity of the Wasserstein metric is  $\mathcal{O}(N^3 \log N)$  for a batch of size  $N$ . Consequently, the complexity of the OvO Wasserstein reaches  $\mathcal{O}(K^2 N^3 \log N)$  for  $K$  clusters. This implies that this GEMINI is hardly usable for a high number of clusters. To tackle this complexity, we propose instead to sample  $M$  independent pairs among the  $K(K-1)/2$  pairs of clusters to compare. We evaluate the OvO Wasserstein on these pairs and scale it to the same order as if performed on all pairs. This is stochastically equivalent to maximising the metric for all pairs:

$$\hat{\mathcal{L}}_{\mathcal{W}}^{\text{ovo}}(\mathbf{x}; y) = \frac{2M}{K(K-1)} \times \sum_{k, k' \in I} p_{\theta}(y=k)p_{\theta}(y=k') \times \mathcal{W}_{\delta} \left( \{m_i^k\}_{i=1}^N, \{m_i^{k'}\}_{i=1}^N \right), \quad (28)$$

with

$$I = \{(k_n, k'_n)\}_{n=1}^M \quad ; (k_n, k'_n) \sim \mathcal{U}(\{1, \dots, K\}^2),$$

a set of uniformly drawn pairs of clusters.

This optimisation requires however longer training time as a tradeoff for a controlled complexity of  $\mathcal{O}(MN^3)$ . Note that the same optimisation can be applied to the OvA Wasserstein-GEMINI.

## APPENDIX J ALL PAIR SHORTEST PATHS DISTANCE

Sometimes, using distances such as the  $\ell_2$  may not capture well the shape of manifolds. To do so, we derive a metric using the all pair shortest paths. Simply put, this metric consists in considering the number of closest neighbors that separates two data samples. To compute it, we first use a sub-metric that we note  $d$ , say the  $\ell_2$  norm. This allows us to compute all distances  $d_{ij}$  between every sample  $i$  and  $j$ . From this matrix of sub-distances, we can build a graph adjacency matrix  $W$  following the rules:

$$W_{ij} = \begin{cases} 1 & d_{ij} \leq \epsilon \\ 0 & d_{ij} > \epsilon \end{cases},$$

where  $\epsilon$  is a chosen threshold such that the graph has sparse edges. Our typical choice for  $\epsilon$  is the 5% quantile of all  $d_{ij}$ .

We chose the graph adjacency matrix to be undirected, owing to the symmetry of  $d_{ij}$  and unweighted. Indeed, solving the all-pairs shortest paths involves the Floyd-Warshall algorithm [59], [66] which complexity  $\mathcal{O}(n^3)$  is not affordable when the number of samples  $n$  becomes large. An undirected and unweighted graph leverages performing  $n$  times the breadth-first-search algorithm, yielding a total complexity of  $\mathcal{O}(n^2 + ne)$  where  $e$  is the number of edges. Consequently, setting a good threshold  $\epsilon$  controls the complexity of the shortest paths to finds. Our final distance between two nodes  $i$  and  $j$  is eventually:

$$c_{ij} = \begin{cases} \text{Shortest-path}_n^W(i, j) & \text{if it exists.} \\ n & \text{otherwise} \end{cases}. \quad (29)$$

This metric  $c$  can then be incorporated inside the Wasserstein-GEMINI.

## APPENDIX K

### PACKAGES FOR EXPERIMENTS

For the implementation details, we use several packages with a python 3.8 version.

- We use PyTorch [54] for all deep learning models and automatic differentiation, as well as NumPy [30] for arrays handling.
- We use Python Optimal Transport's function `emd2` [20] to compute the Wasserstein distances between weighted sums of Diracs.
- We used the implementation of SIMCLR from PyTorch Lightning [19].
- Small datasets such as isotropic Gaussian Mixture of score computations are performed using scikit-learn [55].