f-Divergence Based Classification: Beyond the Use of Cross-Entropy

Nicola Novello [1]   **Andrea M. Tonello** [1]

Abstract

In deep learning, classification tasks are formalized as optimization problems often solved via the minimization of the cross-entropy. However, recent advancements in the design of objective functions allow the usage of the f-divergence to generalize the formulation of the optimization problem for classification. We adopt a Bayesian perspective and formulate the classification task as a maximum a posteriori probability problem. We propose a class of objective functions based on the variational representation of the f-divergence. Furthermore, driven by the challenge of improving the state-of-the-art approach, we propose a bottom-up method that leads us to the formulation of an objective function corresponding to a novel f-divergence referred to as shifted log (SL). We theoretically analyze the objective functions proposed and numerically test them in three application scenarios: toy examples, image datasets, and signal detection/decoding problems. The analyzed scenarios demonstrate the effectiveness of the proposed approach and that the SL divergence achieves the highest classification accuracy in almost all the considered cases.

1. Introduction

Classification problems are relevant in a multitude of domains, such as computer vision, biomedical, and telecommunications engineering (Peng et al., 2010; Nachmani et al., 2018; Uy et al., 2019). In general, classification refers to the estimation of a discrete vector $\mathbf{x}$ (i.e., the class) given an observation vector $\mathbf{y}$. In the Bayesian framework, the optimal method to solve classification problems is derived from the maximum a posteriori probability (MAP) principle (Boyd & Vandenberghe, 2004; Jeong et al., 2024). Classical estimation theory uses a model-based approach to derive the estimation algorithm. Then, the MAP algorithm is well-defined and applicable when the posterior density is known. If the posterior probability is unknown, the first fundamental step towards solving the MAP problem consists of learning the posterior density from the data. In this direction, deep learning (DL) approaches learn the model and solve the estimation task directly via data observation. This is achieved by leveraging artificial neural networks, whose ability to model probability density functions (pdfs) makes them particularly suited for this task, as shown in (Hornik et al., 1989; LeCun et al., 2015; Mohamed & Lakshminarayanan, 2016; Papamakarios et al., 2017). DL models require the design of two main elements: the network architecture, which defines the class of functions the network can estimate, and the objective function that is exploited during the training phase to learn the optimal parameters of the network. In reference to neural network-based classification techniques, most of the previous work focused on the conceievement of the network architecture (Simonyan & Zisserman, 2015; Szegedy et al., 2015; Rezende et al., 2017; Tong et al., 2020; Bhojanapalli et al., 2021). Contrarily, a smaller part of classification literature is dedicated to the objective function design. In most cases, classification is achieved through the minimization of the cross-entropy loss function between the data empirical probability density function p_{data} and the probability density function output of the neural network p_{model} (Kussul et al., 2017; Yang et al., 2018). The minimization of the cross-entropy corresponds to the minimization of the Kullback-Leibler (KL) divergence between the same two probability distributions. The KL divergence is defined as

$$D_{KL}(p_{data}||p_{model}) = \mathbb{E}_{p_{data}}\left[\log\left(\frac{p_{data}}{p_{model}}\right)\right], \quad (1)$$

while the cross-entropy (CE) is $\mathbb{E}_{p_{data}}\left[-\log\left(p_{model}\right)\right]$. Alternatively, some papers propose the usage of proper losses, which is a family of losses characterized by Bregman divergences (Gneiting & Raftery, 2007). For instance, the authors in (Hui & Belkin, 2020) compare the square loss and the cross-entropy for supervised classification tasks, while in (Dong et al., 2019), the authors propose two novel objective functions based on logistic regression. Another popular class of divergence functions is the f-divergence, which has been used for various classification algorithms. The authors in (Yu et al., 2020) propose a min-max game for the objective function design of deep

[1]Department of Networked and Embedded Systems, University of Klagenfurt, Klagenfurt, Austria. Correspondence to: Nicola Novello <nicola.novello@aau.at>.

energy-based models, where they substitute the minimization of the KL with any f-divergence. In (Wei & Liu, 2021), the authors propose a max-max optimization problem to tackle classification with noisy labels. They maximize the f-mutual information (a generalization of the mutual information) between the classifier's output and the true label distribution. In (Zhong & Tandon, 2023), the f-divergence is used as a regularization term in a min-max optimization problem, for the design of fair classifiers (i.e., minimize the classifier discrepancy over sub-groups of the population).

In this paper, we propose to estimate the conditional posterior probability (needed in the MAP classifier) by expressing it as a density ratio (see (9)) and by using a discriminative learning approach (Song & Ermon, 2020). Density ratio estimation approaches have been used in a wide variety of applications (Nowozin et al., 2016; Wei & Liu, 2021; Letizia et al., 2023a). However, unlike known classification objective functions, the main idea behind the proposed estimators is not to use a divergence minimization-based technique. Instead, to solve the classification task, we estimate the ratio between joint and marginal probability densities with a discriminator network and maximize such a ratio (corresponding to the posterior probability) with respect to the class elements. Therefore, contrarily to other f-divergence-based approaches that need a double training optimization procedure (e.g., max-max or min-max), our approach relies on a single training maximization formulation needed to learn the posterior density. Additionally, it enables the use of the f-divergence to obtain a broader set of classifiers beyond the conventional approach based on the exploitation of the cross-entropy.

In more detail, the contributions of this paper are fourfold: a) we design a class of posterior probability estimators that exploits the variational representation of the f-divergence (Nguyen et al., 2010); b) we present a second class of posterior probability estimators formulated using a bottom-up approach; c) we propose a new objective function for classification tasks that corresponds to the variational representation of a novel f-divergence. The proposed divergence is analyzed theoretically and experimentally, and a comparison with the f-divergences known in the literature follows, showing that the new one achieves the best performance in almost all the considered scenarios; d) finally, we conceive two specific network architectures that are trained with the proposed objective functions.

2. MAP-based Classification Through Posterior Probability Learning

In this section, we describe the approach that we propose to tackle classification tasks and the notation used. Let X and Y be two random vectors described by the probability density functions $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Let X
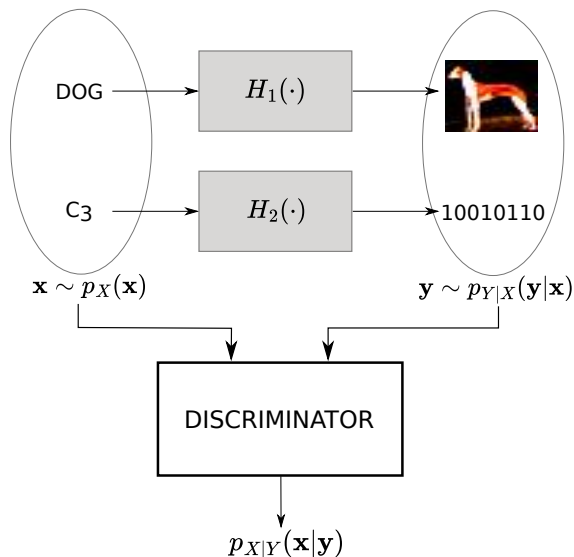

Figure 1. System model representation. X is the input of a stochastic model $H(\cdot)$, while the output is Y. In the example represented by $H_1(\cdot)$, the input is the class element "dog" and the output is an image of a dog. Differently, $H_2(\cdot)$ is a communication channel, therefore the input is a codeword, and the output is the binary representation of such a codeword after the noise addition.

and Y be the input and output of a stochastic model (referred to as $H(\cdot)$), respectively, as shown in Fig. 1. In a classification context, X is discrete and represents the class type with alphabet $\mathcal{A}_x$, while Y is the observation of the class elements.

The MAP estimator (classifier) is formulated as:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x} \in \mathcal{A}_x} p_{X|Y}(\mathbf{x}|\mathbf{y}), \tag{2}$$

where $p_{X|Y}(\mathbf{x}|\mathbf{y})$ is the posterior probability density. In this paper, we propose to adopt a discriminative formulation. Therefore, we express $p_{X|Y}(\mathbf{x}|\mathbf{y})$ as the ratio between $p_{XY}(\mathbf{x}, \mathbf{y})$ and $p_Y(\mathbf{y})$ by using the Bayes theorem.

Since the classification task is an estimation problem, we formulate it with the MAP approach because it is the optimal approach for estimation problems in the Bayesian framework (Proakis & Salehi, 2007; Jeong et al., 2024). A key advantage of the proposed MAP-based approach is that it does not need a double training optimization procedure because it directly learns the posterior pdf formulating the objective function either using the f-divergence (Section 3) or with a bottom-up approach (Section 4). When the posterior pdf is learned, the estimation (thus the classification) problem is solved, because the argmax operator in (2) can be easily computed. First, we study the more general problem of estimating the posterior pdf when X is a continuous random vector. Then, we consider the specific case of X being discrete. The classification problem is solved choosing the optimal class $\mathbf{x}$ that, during inference, maximizes the

estimate of $p_{X|Y}(\mathbf{x}|\mathbf{y})$ (i.e., that solves (2)).

3. Posterior Probability Learning Through the Exploitation of f-Divergence

The first approach we propose is to estimate $p_{X|Y}(\mathbf{x}|\mathbf{y})$ by exploiting the variational representation of the f-divergence.

3.1. f-Divergence

Given two probability distributions P and Q admitting, respectively, the absolutely continuous density functions p and q with respect to $d\mathbf{x}$ defined on the domain $\mathcal{X}$, the f-divergence (also known as Ali-Silvey distance) is defined as (Ali & Silvey, 1966; Csiszár, 1967)

$$D_f(P||Q) = \int_{\mathcal{X}} q(\mathbf{x}) f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x}, \qquad (3)$$

where the *generator function* $f : \mathbb{R}_+ \longrightarrow \mathbb{R}$ is a convex, lower-semicontinuous function such that $f(1) = 0$. Every generator function has a *Fenchel conjugate* function f^*, that is defined as

$$f^*(t) = \sup_{u \in dom_f} \{ut - f(u)\}, \qquad (4)$$

where dom_f is the domain of $f(\cdot)$, f^* is convex and such that $f^{**}(u) = f(u)$.

Leveraging (4), the authors in (Nguyen et al., 2010) expressed a lower bound on any f-divergence, that is referred to as variational representation of the f-divergence:

$$D_f(P||Q) \geq \sup_{T \in \mathcal{T}} \left\{ \mathbb{E}_{p(\mathbf{x})}[T(\mathbf{x})] - \mathbb{E}_{q(\mathbf{x})}[f^*(T(\mathbf{x}))] \right\}, \qquad (5)$$

where $T(\mathbf{x})$ is parametrized by an artificial neural network, and the bound in (5) is tight when $T(\mathbf{x})$ is

$$T^\diamond(\mathbf{x}) = f'\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right), \qquad (6)$$

where f' is the first derivative of f.

3.2. Posterior Estimation Through the Variational Representation of the f-Divergence

Theorem 3.1 provides a class of objective functions that, when maximized, leads to the estimation of the posterior probability density.

Theorem 3.1. *Let X and Y be the random vectors with pdfs $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Assume $\mathbf{y} = H(\mathbf{x})$, where $H(\cdot)$ is a stochastic function, then $p_{XY}(\mathbf{x}, \mathbf{y})$ is the joint density. Define $\mathcal{T}_x$ to be the support of X and $p_U(\mathbf{x})$ to be a uniform pdf with support $\mathcal{T}_x$. Let $f_u : \mathbb{R}_+ \longrightarrow \mathbb{R}$ be a convex function such that $f_u(1) = 0$, and f_u^* be the Fenchel*

conjugate of f_u. Let $\mathcal{J}_f(T)$ be the objective function defined as

$$\mathcal{J}_f(T) = \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim p_{XY}(\mathbf{x},\mathbf{y})}[T(\mathbf{x}, \mathbf{y})]$$
$$- \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim p_U(\mathbf{x}) p_Y(\mathbf{y})}[f_u^*(T(\mathbf{x}, \mathbf{y}))]. \qquad (7)$$

Then,

$$T^\diamond(\mathbf{x}, \mathbf{y}) = \arg\max_{T \in \mathcal{T}} \mathcal{J}_f(T) \qquad (8)$$

leads to the estimation of the posterior density

$$\hat{p}_{X|Y}(\mathbf{x}|\mathbf{y}) = \frac{p_{XY}(\mathbf{x}, \mathbf{y})}{p_Y(\mathbf{y})} = \frac{(f_u^*)'(T^\diamond(\mathbf{x}, \mathbf{y}))}{|\mathcal{T}_x|}, \qquad (9)$$

where $T^\diamond(\mathbf{x}, \mathbf{y})$ is parametrized by an artificial neural network.

According to Theorem 3.1, there exists a class of objective functions to train a discriminator whose output is processed to obtain an estimate of the posterior probability. The choice of the f-divergence offers a degree of freedom (DOF) in the objective function design. To improve the training convergence of the objective functions formulated as variational representation of f-divergences, the literature exploits a change of variable $T(\mathbf{x}, \mathbf{y}) = r(D(\mathbf{x}, \mathbf{y}))$ in (7) (Nowozin et al., 2016). With the same goal, we propose to introduce a second DOF in the development of the objective function (7). Accordingly, it is sufficient to substitute $D^\diamond(\mathbf{x}, \mathbf{y}) = r^{-1}(T^\diamond(\mathbf{x}, \mathbf{y}))$ in (9) to attain the corresponding posterior probability estimator. The exploitation of the DOFs uniquely defines the objective function and numerically impacts the discriminator's parameters convergence during the training phase. We use Theorem 3.1 to obtain five specific estimators trained with the objective functions reported in Tab. 1. These objective functions are obtained by first selecting the generator function and then choosing $r(\cdot)$. The generator functions used to derive Tab. 1 are reported in Tab. 2 of Appendix A and are retrieved from a modification of the well known f-divergences reported for completeness in Tab. 3 of Appendix A. In particular, the modification applied is attained as $f_u^*(t) = |\mathcal{T}_x| f^*(t)$, where $f_u^*(t)$ and $f^*(t)$ are referred to as *unsupervised* and *supervised* generator functions, respectively. In addition, Tab. 1 comprises the change of variable $r(D^\diamond(\mathbf{x}, \mathbf{y}))$ and the expression of $D^\diamond(\mathbf{x}, \mathbf{y})$ for each objective function.

Lemma 3.2 proves the convergence property of any posterior probability estimator that is formulated as in Theorem 3.1.

Lemma 3.2. *Let the artificial neural network $D(\cdot) \in \mathcal{D}$ be with enough capacity and training time (i.e., in the non-parametric limit). Assume the gradient ascent update rule $D^{(i+1)} = D^{(i)} + \mu \nabla \mathcal{J}_f(D^{(i)})$ converges to*

$$D^\diamond = \arg\max_{D \in \mathcal{D}} \mathcal{J}_f(D), \qquad (10)$$

Table 1. Objective functions table. The corresponding f-divergences are: Kullback-Leibler, Reverse Kullback-Leibler, squared Hellinger distance, GAN, and Pearson χ^2.

NAME	OBJECTIVE FUNCTION	$T^\diamond(\mathbf{x}, \mathbf{y})$	$D^\diamond(\mathbf{x}, \mathbf{y})$
$\mathcal{J}_{KL}(D)$	$\mathbb{E}_{p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log(D(\mathbf{x},\mathbf{y}))\Big] - \mathbb{E}_{p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|D(\mathbf{x},\mathbf{y})\Big]$	$\log(D^\diamond(\mathbf{x},\mathbf{y}))+1$	$\frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_Y(\mathbf{y})}$
$\mathcal{J}_{RKL}(D)$	$\mathbb{E}_{p_{XY}(\mathbf{x},\mathbf{y})}\Big[-D(\mathbf{x},\mathbf{y})\Big] + \mathbb{E}_{p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|\log(D(\mathbf{x},\mathbf{y}))\Big]$	$-D^\diamond(\mathbf{x},\mathbf{y})$	$\frac{p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})}$
$\mathcal{J}_{HD}(D)$	$\mathbb{E}_{p_{XY}(\mathbf{x},\mathbf{y})}\Big[-\sqrt{D(\mathbf{x},\mathbf{y})}\Big] - \mathbb{E}_{p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|\frac{1}{\sqrt{D(\mathbf{x},\mathbf{y})}}\Big]$	$1-\sqrt{D^\diamond}$	$\frac{p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})}$
$\mathcal{J}_{GAN}(D)$	$\mathbb{E}_{p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log(1-D(\mathbf{x},\mathbf{y}))\Big] + \mathbb{E}_{p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|\log(D(\mathbf{x},\mathbf{y}))\Big]$	$\log(1-D^\diamond(\mathbf{x},\mathbf{y}))$	$\frac{p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})+p_Y(\mathbf{y})}$
$\mathcal{J}_P(D)$	$\mathbb{E}_{p_{XY}(\mathbf{x},\mathbf{y})}\Big[2(D(\mathbf{x},\mathbf{y})-1)\Big] - \mathbb{E}_{p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|D^2(\mathbf{x},\mathbf{y})\Big]$	$2(D^\diamond(\mathbf{x},\mathbf{y})-1)$	$\frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_Y(\mathbf{y})}$

where $\mathcal{J}_f(D)$ is defined as in (7), *with the change of variable $D = r^{-1}(T)$. Then, the difference between the optimal posterior probability $p^\diamond$ and its estimate $p^{(i)}$ at iteration i is*

$$p^\diamond - p^{(i)} \simeq \frac{1}{|\mathcal{T}_x|}\Big(\delta^{(i)}\Big[(f_u^*)''(r(D^{(i)}))\Big]\Big), \quad (11)$$

where $\delta^{(i)} = r(D^\diamond) - r(D^{(i)})$, $(f_u^*)''$ is the second derivative of f_u^*, and $\mu > 0$ the learning rate. If $D^\diamond$ corresponds to the global optimum achieved by using the gradient ascent method, the posterior probability estimator in (9) *converges to the real value of the posterior density.*

In addition to proving the convergence property of the posterior estimators in Theorem 3.1, Lemma 3.2 provides an intuitive explanation of the posterior probability estimator's bias's dependency on f in (11).

Theorem 3.1 provides an effective method to solve classification problems by designing the objective function based on the choice of an f-divergence. This differs from other methods that leverage f-divergences and that need a dual optimization strategy (Wei & Liu, 2021; Zhong & Tandon, 2023). The proposed method relies on a single optimization problem. In fact, it generalizes the cross-entropy minimization approach (see Section 6). In the next section, we propose a novel bottom-up approach that guides the design of new objective functions for the posterior probability estimation problem.

4. Bottom-Up Posterior Probability Learning

In this section, we propose a bottom-up methodology for developing objective functions that, when maximized, lead to the estimation of the posterior probability. The bottom-up approach reverses the top-down procedure typical of f-divergence formulations (Nowozin et al., 2016; Wei & Liu, 2021). The main advantage of this new method is that it guides the design of the objective function by starting with the imposition of the optimal convergence condition of the discriminator's output $D^\diamond(\mathbf{x}, \mathbf{y})$ (see Appendix C). Theorem 4.1 presents the class of objective functions that, when maximized, leads to the bottom-up posterior estimator.

Theorem 4.1. *Let X and Y be the random vectors with pdfs $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Assume $\mathbf{y} = H(\mathbf{x})$, where $H(\cdot)$ is a stochastic function, then $p_{XY}(\mathbf{x}, \mathbf{y})$ is the joint density. Let $\mathcal{T}_x$ and $\mathcal{T}_y$ be the support of $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Let the discriminator $D(\mathbf{x}, \mathbf{y})$ be a scalar function of $\mathbf{x}$ and $\mathbf{y}$. Let $k(\cdot)$ be any deterministic and invertible function. Then, the posterior density is estimated as*

$$\hat{p}_{X|Y}(\mathbf{x}|\mathbf{y}) = k^{-1}(D^\diamond(\mathbf{x}, \mathbf{y})), \quad (12)$$

where $D^\diamond(\mathbf{x}, \mathbf{y})$ is the optimal discriminator obtained by maximizing

$$\mathcal{J}(D) = \int_{\mathcal{T}_x} \int_{\mathcal{T}_y} \tilde{\mathcal{J}}(D) d\mathbf{x} d\mathbf{y}, \quad (13)$$

for all concave functions $\tilde{\mathcal{J}}(D)$ such that their first derivative is

$$\frac{\partial \tilde{\mathcal{J}}(D)}{\partial D} = \Big(D(\mathbf{x}, \mathbf{y}) - k(p_{X|Y}(\mathbf{x}|\mathbf{y}))\Big)g_1(D, k)$$
$$\triangleq g(D, k) \quad (14)$$

with $g_1(D, k) \neq 0$ deterministic and $\frac{\partial g(D,k)}{\partial D} \leq 0$.

The proposed estimator leverages a discriminative formulation to estimate the density ratio corresponding to the posterior probability. Theorem 4.1 comprises two DOFs in the objective function design. The former is the choice of $k(\cdot)$, since the discriminator estimates an invertible transformation of the posterior density. We noticed that the classifier performs better when $k(\cdot)$ is chosen to resemble particular activation functions. The second DOF, represented by $g_1(\cdot)$, is a rearrangement term that modifies the result of the integration of $g(\cdot)$ with respect to D. The exploitation of such a DOF allows to attain different objective functions even when imposing the same optimal discriminator's output. Theorems 3.1 and 4.1 have different advantages. The benefit of Theorem 3.1 is its simple applicability, that must, however, begin with the selection of the generator function f. Theorem 4.1 has the advantage of guiding the objective function design without relying on existing f-divergences. Corollary 4.2 in Appendix B.6 exploits the DOFs in the

choice of $k(\cdot)$ and $g_1(\cdot)$ to obtain the objective functions in Tab. 1, which were previously derived from Theorem 3.1. An interesting result highlighted by Corollary 4.2 is that all the objective functions corresponding to the well-known f-divergences in Tab. 2 use the same class of functions $g_1(\cdot) \propto 1/D^\alpha \cdot 1/(1-D)^\beta$ expressed in (64) in Appendix B.6. This property, highlighted by the class of bottom-up estimators, shows us that $\mathcal{J}_{GAN}(D)$ is the only one using $\beta \neq 0$. Thanks to the observation of this peculiarity and driven by curiosity, we develop a new objective function starting from $k_{GAN}(\cdot)$ and $g_{1,GAN}(\cdot)$ but imposing $\beta = 0$.

5. Shifted Log Objective Function and Divergence

In this section, we present a new objective function for classification problems, that we design by using Theorem 4.1. Then, we prove that such an objective function corresponds to the variational representation of a novel f-divergence, called *shifted log* (SL). We will demonstrate in Section 7 that such a new objective function achieves the best performance in almost all the classification scenarios discussed.

Theorem 5.1. *Let X and Y be two random vectors with pdfs $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Assume $Y = H(X)$, with $H(\cdot)$ stochastic function, then let $p_{XY}(\mathbf{x}, \mathbf{y})$ be the joint density. Let $\mathcal{T}_x$ be the support of X. Let $p_U(\mathbf{x})$ be a uniform pdf having the same support $\mathcal{T}_x$. The maximization of the objective function*

$$\mathcal{J}_{SL}(D) = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[D(\mathbf{x},\mathbf{y})\Big]$$
$$+ \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|\Big(\log(D(\mathbf{x},\mathbf{y})) - D(\mathbf{x},\mathbf{y})\Big)\Big],$$
(15)

leads to the optimal discriminator output

$$D^\diamond(\mathbf{x}, \mathbf{y}) = \arg\max_D \mathcal{J}_{SL}(D) = \frac{1}{1 + p_{X|Y}(\mathbf{x}, \mathbf{y})}, \quad (16)$$

and the posterior density estimate is computed as

$$\hat{p}_{X|Y}(\mathbf{x}|\mathbf{y}) = \frac{1 - D^\diamond(\mathbf{x}, \mathbf{y})}{D^\diamond(\mathbf{x}, \mathbf{y})}. \quad (17)$$

Corollary 5.2 states that $\mathcal{J}_{SL}(D)$ in (15) can be obtained from Theorem 3.1 by using a new f-divergence referred to as shifted log.

Corollary 5.2. *Define the generator function of the shifted log divergence*

$$f_{u,SL}(u) = -|\mathcal{T}_x|\log(u + |\mathcal{T}_x|) + K, \quad (18)$$

where $K = |\mathcal{T}_x|\log(1 + |\mathcal{T}_x|)$ is constant. Then, $\mathcal{J}_{SL}(D)$ in (15) is the variational representation of $D_{f_{u,SL}}(p_{XY}\|p_U p_Y)$.

5.1. Remarks on the New Objective Function and *f*-Divergence

Since the proposed objective function in (15) is the variational representation of an f-divergence (for Corollary 5.2), Lemma 3.2 ensures the convergence property of its estimate to the true posterior probability in the nonparametric limit.

The supervised version of the SL divergence is

$$f_{SL}(u) = -\log(u + 1) + \log(2), \quad (19)$$

which is obtained by substituting $|\mathcal{T}_x| = 1$ (see Appendix A). Although f_{SL} is obtained in the context of posterior estimation problems, the proposed f-divergence can be applied to a broader variety of tasks. Since D_{SL} is upper-bounded (see Corollary 5.3 in Appendix B.9), it is a suitable generator function when the optimization problem requires to maximize the variational representation of the f-divergence (i.e., a max-max game). For instance, for classification tasks as in (Wei & Liu, 2021) or for representation learning applications as in (Hjelm et al., 2019).

5.2. Comparison Between SL and GAN Divergences

The GAN divergence is known to be highly-performing in a wide variety of tasks (Nowozin et al., 2016; Hjelm et al., 2019). Moreover, the objective functions corresponding to SL and GAN divergences can be obtained from Theorem 4.1 by choosing the same $k(\cdot)$, but different $g_1(\cdot)$. Corollary 5.4 compares the concavity of the two objective functions in the neighborhood of the global optimum.

Corollary 5.4. *Let $\mathcal{J}_{SL}(D)$ be defined as in (15). Let $\mathcal{J}_{GAN}(D)$ be defined as in Tab. 1. Let $D_N^\diamond$ be the discriminator output in a neighborhood of $D^\diamond$ where $\mathcal{J}_{SL}(D_N^\diamond)$ and $\mathcal{J}_{GAN}(D_N^\diamond)$ are concave. Then,*

$$\left|\frac{\partial \mathcal{J}_{GAN}(D_N^\diamond)}{\partial D}\right| \geq \left|\frac{\partial \mathcal{J}_{SL}(D_N^\diamond)}{\partial D}\right|. \quad (20)$$

Although the shape of the loss landscape depends on many factors (e.g., the batch size (Keskar et al., 2017; Chaudhari et al., 2019)), sharper maxima attain larger test error (Li et al., 2018). The steepness of the concavity of the objective function in the neighborhood of the global optimum provides insights on the basin of attraction of the point of maximum. Intuitively, a flatter landscape corresponds to a larger basin of attraction of the global optimum, rendering training with the SL divergence better than with the GAN divergence. The results in Fig. 5 and Tab. 6 in Appendix D validate the findings of Corollary 5.4.

6. Discriminator Architecture

In this section, we discuss the appropriate modifications to the discriminator's architecture to suit our estimators to

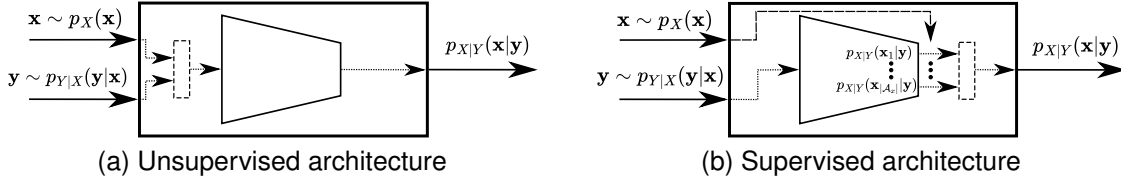(a) Unsupervised architecture

(b) Supervised architecture

Figure 2. Diagrams of unsupervised and supervised architectures. The thick rectangle delineates the *discriminator* in Fig. 1. The trapezoidal shape represents the neural network architecture.

the classification scenario, where the number of classes is finite. The architecture type used differs depending on the alphabet of X, referred to as $\mathcal{A}_x$. When X is continuous, we use a structure referred to as *unsupervised architecture*. Differently, when X is discrete, we use a structure referred to as *supervised architecture*.

6.1. Unsupervised Architecture

In this setting, the samples $\mathbf{x}$ and $\mathbf{y}$ drawn from the empirical probability distributions $p_{XY}(\mathbf{x}, \mathbf{y})$ and $p_U(\mathbf{x})p_Y(\mathbf{y})$ are concatenated and fed into the discriminator. The discriminator output is a scalar, since it is the posterior density function estimate corresponding to the pair $(\mathbf{x}, \mathbf{y})$ given as input. The discriminator architecture is represented in Fig. 2(a), where the concatenation between the $\mathbf{x}$ and $\mathbf{y}$ realizations is identified by a dashed rectangle.

6.2. Supervised Architecture

The supervised architecture (Fig. 2(b)) introduces in the problem's formulation the constraint that $\mathcal{A}_x$ is a set containing a finite number of elements $\mathcal{A}_x = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. This constraint is embedded in the architecture (highlighted by a dashed arrow in Fig. 2(b)), so that the output layer contains one neuron for each sample in $\mathcal{A}_x$. With this modification, the i-th output neuron returns $p_{X|Y}(\mathbf{x}_i|\mathbf{y})$. Accordingly, the input layer is fed with only the realizations $\mathbf{y}$. Notation wise, the discriminator's output is referred to as $\mathbf{D}(\mathbf{y})$. Theorem 6.1 shows the modification to the formulation of the objective functions in (7) when using a supervised architecture. First, we define the notation useful for the theorem statement: $\mathbf{D}(\mathbf{y}) = [D(\mathbf{x}_1, \mathbf{y}), \ldots, D(\mathbf{x}_m, \mathbf{y})]$ and $\mathbf{1}_m(\mathbf{x}_i) = [0, \ldots, 0, \underbrace{1}_{i^{th}\ pos.}, 0, \ldots, 0]^T$.

Theorem 6.1. *Let $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$ be pdfs describing the input and output of a stochastic function $H(\cdot)$, respectively. Let $p_X(\mathbf{x}) \triangleq \sum_{i=1}^m P_X(\mathbf{x}_i)\delta(\mathbf{x} - \mathbf{x}_i)$, where $P_X(\cdot)$ is the probability mass function of X. Let $\mathcal{T}_x$ be the support of $p_X(\mathbf{x})$ and $|\mathcal{T}_x|$ its Lebesgue measure. Let $p_U(\mathbf{x})$ be the uniform discrete pdf over $\mathcal{T}_x$. Let the discriminator be characterized by a supervised architecture. Then, the objective*

function in (7) *becomes*

$$\mathcal{J}(D) = \mathbb{E}_{\mathbf{x} \sim p_X(\mathbf{x})}\left[\mathbb{E}_{\mathbf{y} \sim p_{Y|X}(\mathbf{y}|\mathbf{x})}\left[r(\mathbf{D}(\mathbf{y}))\mathbf{I}_m(\mathbf{x})\right]\right]$$
$$- \mathbb{E}_{\mathbf{y} \sim p_Y(\mathbf{y})}\left[\sum_{i=1}^m f^*\left(r(D(\mathbf{x}_i, \mathbf{y}))\right)\right], \quad (21)$$

where $D(\mathbf{x}_i, \mathbf{y})$ is the i-th component of $\mathbf{D}(\mathbf{y})$ and $\mathbf{T}(\mathbf{y}) = r(\mathbf{D}(\mathbf{y}))$.

The supervised versions of the objective functions utilized in this paper are listed in Section A of the Appendix. For classification problems, the objective function obtained by substituting the KL divergence in Theorem 6.1 (see (25) in Appendix A.1) is exactly the cross-entropy loss. When the Softmax function is applied to the discriminator output (because $D^\diamond(\mathbf{x}, \mathbf{y}) = p_{X|Y}(\mathbf{x}|\mathbf{y})$ is a discrete pdf), the expectation over $p_Y(\mathbf{y})$ becomes a constant always equal to 1. Maximizing (25) is equivalent to minimizing the negative of (25), which is precisely the minimization of the CE.

7. Results

In this section, we report several numerical results to assess the validity of the methods proposed to estimate the posterior probability density and enable the classification task. The considered scenarios are: classification for image datasets; signal decoding in telecommunications engineering cast into a classification task; posterior probability estimation when $p_X(\mathbf{x})$ is continuous. The results demonstrate that different f-divergences attain different performance. We show that the KL divergence (thus the CE) is not necessarily the best choice for classification tasks, and more in general for probability estimation problems. We demonstrate that the SL divergence achieves the best performance in almost all the tested contexts. When referring to the performance of any f-divergence, we will implicitly imply the performance achieved using the objective function derived using such an f-divergence.

The first two scenarios are classification problems, therefore we use the supervised formulation of the discriminator architecture. The third scenario comprises two toy cases, where

6

Table 4. Classification accuracy on MNIST (M), Fashion MNIST (FM), CIFAR10 (C10), and CIFAR100 (C100). The MobileNetV2 is referred to as MobileNet.

DATASET	MODEL	CE	RKL	HD	GAN	P	SL
M	SMALL	**99.08** $\pm$ 0.06	96.05 $\pm$ 0.25	98.68 $\pm$ 0.05	**99.08** $\pm$ 0.07	98.89 $\pm$ 0.08	99.03 $\pm$ 0.04
FM	SMALL	91.64 $\pm$ 0.09	82.63 $\pm$ 1.78	90.75 $\pm$ 0.13	91.63 $\pm$ 0.10	89.86 $\pm$ 0.67	**91.83** $\pm$ 0.02
C10	SMALL	70.13 $\pm$ 0.05	63.59 $\pm$ 0.34	69.38 $\pm$ 0.28	69.98 $\pm$ 0.15	59.62 $\pm$ 0.45	**70.87** $\pm$ 0.26
	VGG	93.69 $\pm$ 0.03	84.24 $\pm$ 2.21	93.51 $\pm$ 0.06	93.75 $\pm$ 0.04	84.79 $\pm$ 0.21	**93.93** $\pm$ 0.08
	DLA	95.04 $\pm$ 0.02	90.83 $\pm$ 0.10	94.56 $\pm$ 0.11	95.04 $\pm$ 0.13	91.61 $\pm$ 0.21	**95.31** $\pm$ 0.09
	RESNET	95.39 $\pm$ 0.04	92.88 $\pm$ 0.26	95.15 $\pm$ 0.08	95.24 $\pm$ 0.06	93.78 $\pm$ 0.21	**95.43** $\pm$ 0.04
	MOBILENET	92.59 $\pm$ 0.13	83.97 $\pm$ 0.21	91.95 $\pm$ 0.33	92.37 $\pm$ 0.14	84.30 $\pm$ 0.32	**93.89** $\pm$ 0.15
C100	VGG	72.73 $\pm$ 0.30	45.80 $\pm$ 2.86	73.51 $\pm$ 0.03	68.88 $\pm$ 0.20	37.19 $\pm$ 0.66	**73.61** $\pm$ 0.05
	DLA	76.29 $\pm$ 0.43	68.86 $\pm$ 1.17	78.63 $\pm$ 0.14	77.34 $\pm$ 0.22	57.97 $\pm$ 0.07	**78.65** $\pm$ 0.01
	RESNET	**78.29** $\pm$ 0.18	70.68 $\pm$ 0.44	77.59 $\pm$ 0.06	77.43 $\pm$ 0.08	61.12 $\pm$ 0.23	78.03 $\pm$ 0.04
	MOBILENET	72.61 $\pm$ 0.08	53.17 $\pm$ 0.35	73.00 $\pm$ 0.30	65.66 $\pm$ 0.46	46.00 $\pm$ 0.37	**74.78** $\pm$ 0.23

we show that the unsupervised formulation of the proposed estimators works for continuous random vectors X. Before discussing the numerical results, we briefly describe the details of the code implementation[1].

7.1. Implementation Details

Supervised Architecture: For the first scenario (Section 7.2), we use convolutional neural networks. When referring to *small* network, we use a discriminator comprising a small set of convolutional layers (less than 4) followed by a feedforward fully connected part. Besides the small network, in the first scenario we utilize various deep network architectures: VGG (Simonyan & Zisserman, 2015), ResNet (He et al., 2016b), DLA (Yu et al., 2018), and MobileNetV2 (Sandler et al., 2018). The discriminator hyper-parameters slightly vary depending on the dataset tested. The network parameters are updated by using SGD with momentum. The activation function of the last layer depends on the objective function optimized during the training phase.

For the second scenario (Section 7.3), we use fully connected feedforward neural networks. The architecture used for the decoding scenario comprises two hidden layers with 100 neurons each. The network weights are updated by using the Adam optimizer (Kingma & Ba, 2015). The LeakyReLU activation function is utilized in all the layers except the last one, where the activation function is chosen based on the objective function. In some cases, the Dropout technique (Srivastava et al., 2014) helps the convergence of the training process.

Unsupervised Architecture: The discriminator architecture utilized for the unsupervised tasks comprises two hidden layers with 100 neurons each and the LeakyReLU activation function. The activation function of the output layer depends on the objective function used during training. The network weights and biases are updated by using the Adam

---

[1]Our implementation can be found at https://github.com/tonellolab/discriminative-classification-fDiv

optimizer. Dropout is used during training.

7.2. Image Datasets Classification

The first scenario tackled is the classification of image datasets. The objective functions performance is tested for the MNIST (LeCun et al., 1998), Fashion MNIST (Xiao et al., 2017), CIFAR10, and CIFAR100 (Krizhevsky et al., 2009) datasets. A more detailed description of the datasets can be found in Appendix D. We compare the classification accuracy of the supervised versions of the objective functions in Tab. 1 and in (15), which are all reported in Appendix A. To improve the training procedure, we apply data augmentation on the CIFAR datasets by randomly cropping and flipping the images. The learning rate is initially set to 0.1 and then we use a cosine annealing scheduler (Loshchilov & Hutter, 2017) to modify its value during the 200 epochs of training. We compute the mean accuracy
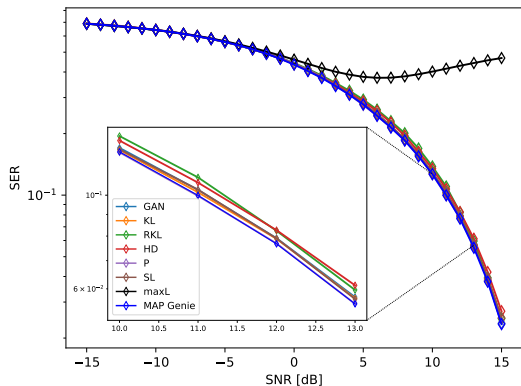

Figure 3. SER achieved by using a 4-PAM modulation over a non-linear communication channel.

and its standard deviation for each dataset and f-divergence by running the code over multiple random seeds. The classification accuracy results displayed in Tab. 4 (and in the extended version comprising more network architectures in Tab. 5 in Appendix D) confirm that each generator function
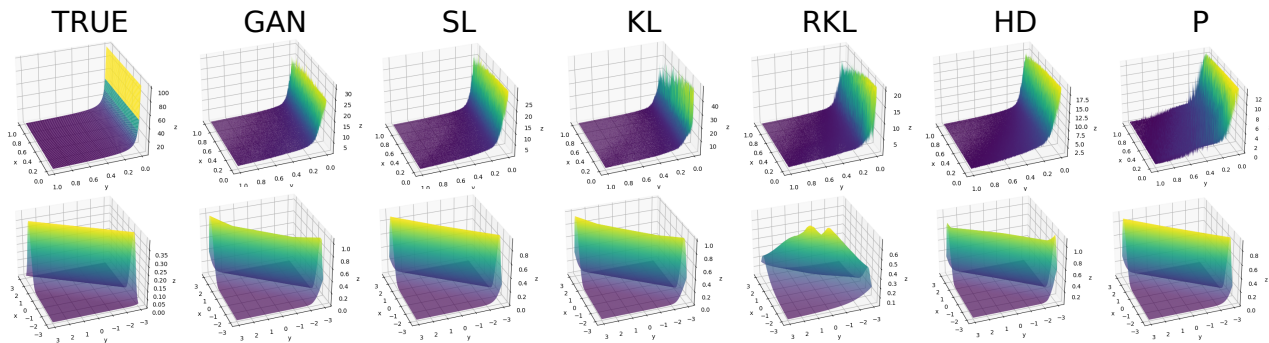
Figure 4. Continuous posterior density estimation for various f-divergences. The results of the Exponential task is represented in the upper row, while the outcomes of the Gaussian task are depicted in the lower row. The true posterior density is the first plot of each row.

has a different impact on the neural network training, as also shown in (Nowozin et al., 2016; Wei & Liu, 2021). The SL divergence attains the highest classification accuracy for almost all the architectures tested, showing its effectiveness compared to the other divergences and its stable behavior over different datasets and architectures. In particular, the SL divergence attains better performance than the CE, which corresponds to the state-of-the-art approach for image classification tasks. Furthermore, in the few scenarios where the SL divergence does not achieve the best performance, it achieves the second best (only in one case the third best, in Table 5 in Appendix D) performance, with an accuracy close to the optimal one (see Tables 4, 5). The CE is, on average, the second-best objective function. Conversely, the Pearson χ^2 performs the worst in almost all the scenarios. Moreover, except from the SL divergence and the CE, the performance of the other f-divergences is more architecture dependent. For instance, as reported in Tab. 5 in Appendix D, the PreActResNet (He et al., 2016a) attains the highest accuracy when it is trained with the GAN-based objective function. The choice of the architecture often depends on the goal of the classification algorithm. For embedded systems, light architectures are used. Therefore, the MobileNetV2 is an option (Chiu et al., 2020). In such a case, the SL divergence obtains an accuracy $1.5/2\%$ higher than the CE, which makes the SL the preferred choice for the network's training. Additional numerical results on other network architectures and on the speed of convergence of the training phase are reported in Appendix D for space limitations. The speed of convergence analysis (Fig. 5) demonstrates that the training with the SL divergence leads to a faster convergence to the optimum discriminator w.r.t. the GAN divergence, as stated in Corollary 5.4.

7.3. Signal Decoding

The second scenario is the decoding problem. Decoding a sequence of received bits is crucial in a telecommunication system. In some cases, when the communication channel

is known, the optimal decoding technique is also known (Proakis & Salehi, 2007). However, the communication channel is generally unknown, and DL-based techniques can be used to learn it (Soltani et al., 2019). By knowing that the optimal decoding criterion is the posterior probability maximization, we demonstrate that the proposed MAP approach solves the decoding problem and that the supervised version of $\mathcal{J}_{SL}(D)$ (see (37) in Appendix A.6) achieves optimal performance. We consider a 4-PAM (i.e., pulse amplitude modulation with four symbols) over a nonlinear channel with additive Gaussian noise. In particular, given the symbol at time instant t (referred to as x_t), we obtain the channel output as $y_t = sgn(x_t)\sqrt{|x_t|} + n_t$, where n_t is the Gaussian noise and $sgn(\cdot)$ is the sign function. We show the symbol error rate (SER) behavior achieved by the proposed posterior estimators when varying the signal-to-noise ratio (SNR) in Fig. 3. The estimators proposed are compared with the results of the max likelihood estimator (referred to as maxL) and the MAP Genie estimators. The MAP Genie estimator uses the knowledge of the channel nonlinearity to decode the received sequence of bits. The proposed list of posterior probability estimators performs better than the max likelihood estimator, achieving accuracy close to the optimal MAP Genie estimator. In Appendix D, other scenarios of decoding tasks are reported.

7.4. Continuous Posterior Estimation

This section considers two toy examples for the case $|\Omega| \geq |\mathbb{R}|$. The comparison between the closed-form of the posterior distribution and the discriminator estimate is showed in Fig. 4. The discriminator prediction is obtained by training a small neural network with the objective functions reported in Tab. 1 and in (15). The closed-form posterior distribution (referred to as *true* in Fig. 4) is the first element of each row.

Exponential task. In the first toy example, we define the model $Y = X + N$, where X and N are independent Exponential random variables. Therefore, Y is a Gamma

distribution (Durrett, 2019). The closed-form posterior probability can be expressed as in (106) (see Appendix B.12 for the formula and proof). Similarly to the previous numerical results, different f-divergences lead to diverse estimates of the posterior density. The objective functions corresponding to the GAN, SL, and HD divergences attain better estimates w.r.t. the others. For a fixed y, in fact, the posterior density value is constant over x.

Gaussian task. In the second toy example, we consider the model $Y = X + N$, where X, N are independent Gaussian random variables. Thus, Y is a Gaussian distribution. The posterior density expression is reported in Appendix B.13 (see (117)). The objective functions corresponding to the SL, KL, and P divergences attain better estimates w.r.t. the others, since the estimate attains the desired saddle shape.

8. Conclusions

In this paper, we proposed a new MAP perspective for supervised classification problems. We have proposed to use a discriminative formulation to express the posterior probability density, and we have derived two classes of estimators to estimate it. From them, we extracted a list of posterior probability estimators and compared them with the notorious cross-entropy minimization approach. Numerical results on different scenarios demonstrate the effectiveness of the presented estimators and that the proposed SL divergence achieves the highest classification accuracy in almost all the scenarios. Additionally, we show that the proposed posterior probability estimators work for the general case of continuous a priori probabilities, for which we design a specific neural network architecture.

Acknowledgements

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ali, S. M. and Silvey, S. D. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142, 1966.

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., and Veit, A. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10231–10241, 2021.

Boyd, S. P. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Chiu, Y.-C., Tsai, C.-Y., Ruan, M.-D., Shen, G.-Y., and Lee, T.-T. Mobilenet-ssdv2: An improved object detection model for embedded systems. In *2020 International conference on system science and engineering (ICSSE)*, pp. 1–5. IEEE, 2020.

Csiszár, I. On information-type measure of difference of probability distributions and indirect observations. *Studia Sci. Math. Hungar.*, 2:299–318, 1967.

Dong, Q., Zhu, X., and Gong, S. Single-label multi-class image classification by deep logistic regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3486–3493, 2019.

Durrett, R. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 630–645. Springer, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016b.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations, ICLR*, 2019.

Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017.

Hui, L. and Belkin, M. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. In *International Conference on Learning Representations, ICLR*, 2020.

Jeong, M., Cardone, M., and Dytso, A. Demystifying the optimal performance of multi-class classification. *Advances in Neural Information Processing Systems*, 36, 2024.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations, ICLR*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5):778–782, 2017.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Letizia, N. A., Novello, N., and Tonello, A. M. Variational *f*-divergence and derangements for discriminative mutual information estimation. *arXiv preprint arXiv:2305.20025*, 2023a.

Letizia, N. A., Tonello, A. M., and Poor, H. V. Cooperative channel capacity learning. *IEEE Communications Letters*, 2023b.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations, ICLR*, 2017.

Mohamed, S. and Lakshminarayanan, B. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Nachmani, E., Marciano, E., Lugosch, L., Gross, W. J., Burshtein, D., and Be'ery, Y. Deep learning methods for improved decoding of linear codes. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):119–131, 2018. doi: 10.1109/JSTSP.2017.2788405.

Nguyen, X., Wainwright, M. J., and Jordan, M. I. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010. doi: 10.1109/TIT.2010.2068870.

Nowozin, S., Cseke, B., and Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Peng, Y., Wu, Z., and Jiang, J. A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics*, 43(1):15–23, 2010.

Proakis, J. G. and Salehi, M. *Fundamentals of communication systems*. Pearson Education India, 2007.

Rezende, E., Ruppert, G., Carvalho, T., Ramos, F., and De Geus, P. Malicious software classification using transfer learning of resnet-50 deep neural network. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1011–1014. IEEE, 2017.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Soltani, M., Pourahmadi, V., Mirzaei, A., and Sheikhzadeh, H. Deep learning-based channel estimation. *IEEE Communications Letters*, 23(4):652–655, 2019. doi: 10.1109/LCOMM.2019.2898944.

Song, J. and Ermon, S. Understanding the limitations of variational mutual information estimators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Tonello, A. M. and Letizia, N. A. Mind: Maximum mutual information based neural decoder. *IEEE Communications Letters*, 26(12):2954–2958, 2022.

Tong, W., Chen, W., Han, W., Li, X., and Wang, L. Channel-attention-based densenet network for remote sensing image scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13: 4121–4132, 2020.

Uy, M. A., Pham, Q.-H., Hua, B.-S., Nguyen, T., and Yeung, S.-K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1588–1597, 2019.

Vajda, I. On the f-divergence and singularity of probability measures. *Periodica Mathematica Hungarica*, 2(1-4): 223–234, 1972.

Wei, J. and Liu, Y. When optimizing *f*-divergence is robust with label noise. In *International Conference on Learning Representations, ICLR*, 2021.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Yang, X., Ye, Y., Li, X., Lau, R. Y. K., Zhang, X., and Huang, X. Hyperspectral image classification with deep learning models. *IEEE Transactions on Geoscience and Remote Sensing*, 56(9):5408–5423, 2018. doi: 10.1109/ TGRS.2018.2815613.

Yu, F., Wang, D., Shelhamer, E., and Darrell, T. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2403–2412, 2018.

Yu, L., Song, Y., Song, J., and Ermon, S. Training deep energy-based models with f-divergence minimization. In *International Conference on Machine Learning*, pp. 10957–10967. PMLR, 2020.

Zhong, M. and Tandon, R. Learning fair classifiers via min-max f-divergence regularization. In *2023 59th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1–8. IEEE, 2023.

Zhu, H., Huang, H., Li, Y., Zheng, A., and He, R. Arbitrary talking face generation via attentional audio-visual coherence learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2021.

Table 2. Unsupervised f-divergences table. The corresponding f-divergences are: Kullback-Leibler, Reverse Kullback-Leibler, squared Hellinger distance, GAN, and Pearson χ^2.

NAME	$f_u(u)$	$f_u^*(t)$
KL	$u \log \left(\frac{u}{\|\mathcal{T}_x\|} \right)$	$\|\mathcal{T}_x\| \exp{(t - 1)}$
RKL	$-\|\mathcal{T}_x\| \log(u)$	$-\|\mathcal{T}_x\|(1 + \log(-t))$
HD	$(\sqrt{u} - \sqrt{\|\mathcal{T}_x\|})^2$	$\|\mathcal{T}_x\| \frac{t}{1-t}$
GAN	$u \log(u) - (u + \|\mathcal{T}_x\|) \log(u + \|\mathcal{T}_x\|)$	$-\|\mathcal{T}_x\| \log(1 - \exp{(t)})$
P	$\frac{1}{\|\mathcal{T}_x\|}(u - \|\mathcal{T}_x\|)^2$	$\|\mathcal{T}_x\| \left(\frac{1}{4}t^2 + t \right)$

Table 3. f-divergences table. The corresponding f-divergences are: Kullback-Leibler, Reverse Kullback-Leibler, squared Hellinger distance, GAN, and Pearson χ^2.

NAME	$f(u)$	$f^*(t)$
KL	$u \log(u)$	$\exp{(t - 1)}$
RKL	$-\log(u)$	$-1 - \log(-t)$
HD	$(\sqrt{u} - 1)^2$	$\frac{t}{1-t}$
GAN	$u \log(u) - (u + 1) \log(u + 1)$	$-\log(1 - \exp{(t)})$
P	$(u - 1)^2$	$\frac{1}{4}t^2 + t$

A. Appendix: Objective Functions Used in the Experiments

The unsupervised and supervised versions of the objective functions used to achieve the results showed in Section 7 are reported in this section. The training part consists in the alternation of two phases. In the former phase, the network is fed with N realizations of the joint distribution $p_{XY}(\mathbf{x}, \mathbf{y})$ to compute the first term of the objective function. In the latter phase, the type of architecture defines the procedure to compute the second term of the objective function. The unsupervised architecture is fed with N samples drawn from $p_U(\mathbf{x})p_Y(\mathbf{y})$. The supervised architecture is fed with N samples drawn from $p_Y(\mathbf{y})$ (see Theorem 6.1).

During the test part, the network is fed with the samples drawn from the joint distribution $((\mathbf{x}, \mathbf{y}) \sim p_{XY}(\mathbf{x}, \mathbf{y}))$, and the posterior probability density estimate is obtained as in (9) or (12).

We report the objective functions derived from well-known f-divergences. Therefore, we first report the notorious f-divergences in Table 3, and their unsupervised version in Table 2. These tables do not contain the constant terms that render $f_u(1) = 0$ or $f(1) = 0$, as their presence do not affect the optimization of the derived objective functions (see the proof of Lemma B.2). Let f_u and f_u^* the unsupervised generator function and its Fenchel conjugate in Table 2, respectively. Differently, f and f^* are the supervised generator function and its Fenchel conjugate in Table 3, respectively. Then,

$$f_u^*(t) = |\mathcal{T}_x| f^*(t), \tag{22}$$

where $|\mathcal{T}_x|$ is the Lebesgue measure of the support of $p_X(\mathbf{x})$. Accordingly,

$$f_u(u) = \sup_{t \in dom_{f_u^*}} \{ut - f_u^*(t)\}. \tag{23}$$

Vice versa, the supervised version of f_u and f_u^* can be easily attained by substituting $|\mathcal{T}_x| = 1$ in the unsupervised formulation.

The usage of (22) in Theorem 3.1 is needed because $|\mathcal{T}_x|$ counterbalances the effect of $p_U(\mathbf{x})$ to obtain the pdf ratio equivalent to the posterior density $p_{X|Y}(\mathbf{x}|\mathbf{y}) = p_{XY}(\mathbf{x}, \mathbf{y})/p_Y(\mathbf{y})$. In fact, $|\mathcal{T}_x|$ is included in the expectation term computed over $p_U(\mathbf{x})$. In Lemma B.2, we prove that f_u is the generator function of a valid f-divergence.

Note: The notations *supervised* and *unsupervised* refer to the discriminator architecture. The former one is the typical supervised classification architecture, while the latter one does not use X as the set of labels, but as an additional input. The terminology we use to distinguish between the two different versions of f-divergences (f and f_u) is a consequence of the architecture's notation. The unsupervised architecture, in fact, is trained using objective functions defined with f_u (see Theorem 3.1 and Corollary 5.2), while the supervised architecture is trained by using objective functions based on f (see Theorem 6.1).

We recall the notation used in the objective functions: $\mathbf{D}(\mathbf{y}) = [D(\mathbf{x}_1, \mathbf{y}), \ldots, D(\mathbf{x}_m, \mathbf{y})]$ is a row vector, $\mathbf{1}_m = [1, \ldots, 1]^T$ is a column vector and $\mathbf{1}_m(\mathbf{x}_i) = [0, \ldots, 0, \underbrace{1}_{i^{th}\ pos.}, 0, \ldots, 0]^T$ is a column vector.

A.1. KL-Based Objective Functions

Kullback-Leibler Divergence: The variational representation of the KL divergence is achieved substituting $f^*_{u,KL}$ listed in Table 2 in (7). This leads to the unsupervised objective function

$$\mathcal{J}_{KL}(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log(D(\mathbf{x},\mathbf{y}))\Big] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|D(\mathbf{x},\mathbf{y})\Big] + 1, \tag{24}$$

where $T^\diamond(\mathbf{x},\mathbf{y}) = \log(\frac{p_{XY}(\mathbf{x},\mathbf{y})}{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}) + 1$, $p_U(\mathbf{x}) = \frac{1}{|\mathcal{T}_x|}$, and $D^\diamond(\mathbf{x},\mathbf{y}) = \frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_Y(\mathbf{y})}$. The supervised version of the objective function in (24) is attained by using Theorem 6.1:

$$\mathcal{J}_{KL}(D) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\Big[\mathbb{E}_{\mathbf{y}\sim p_{Y|X}(\mathbf{y}|\mathbf{x})}\Big[\log(\mathbf{D}(\mathbf{y}))\mathbf{1}_m(\mathbf{x})\Big]\Big] - \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\Big[\mathbf{D}(\mathbf{y})\mathbf{1}_m\Big]. \tag{25}$$

When the last layer of the supervised discriminator utilizes the softmax activation function (i.e., when the output is normalized to be a discrete probability density function), then the second term in (25) is always equal to 1. Thus, the maximization of (25) exactly corresponds to the minimization of the KL divergence in (1), and therefore to the minimization of the cross-entropy.

Interestingly, the more general formulation in (25) allows the usage of different activation functions in the last layer, with the only requirement that the discriminator's output is constrained to assume positive values (e.g., softplus).

A.2. RKL-Based Objective Functions

Reverse Kullback-Leibler Divergence: Theorem 3.1 leads to the variational representation of the reverse KL divergence, when substituting $f^*_{u,RKL}$ listed in Table 2 in (7)

$$\mathcal{J}_{RKL}(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[-D(\mathbf{x},\mathbf{y})\Big] + \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|\log(D(\mathbf{x},\mathbf{y}))\Big], \tag{26}$$

where $T^\diamond(\mathbf{x},\mathbf{y}) = -\frac{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})}$ and $D^\diamond(\mathbf{x},\mathbf{y}) = \frac{p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})}$. The supervised version of the objective function in (26) is obtained by using Theorem 6.1.

$$\mathcal{J}_{RKL}(D) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\Big[\mathbb{E}_{\mathbf{y}\sim p_{Y|X}(\mathbf{y}|\mathbf{x})}\Big[-\mathbf{D}(\mathbf{y})\mathbf{1}_m(\mathbf{x})\Big]\Big] + \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\Big[\log(\mathbf{D}(\mathbf{y}))\mathbf{1}_m\Big]. \tag{27}$$

A.3. HD-Based Objective Functions

Hellinger Squared Distance: Theorem 3.1 leads to the variational representation of the Hellinger squared distance, when substituting $f^*_{u,HD}$ listed in Table 2 in (7)

$$\mathcal{J}_{HD}(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[-\sqrt{D(\mathbf{x},\mathbf{y})}\Big] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|\frac{1}{\sqrt{D(\mathbf{x},\mathbf{y})}}\Big], \tag{28}$$

where $T^\diamond(\mathbf{x},\mathbf{y}) = \Big(\sqrt{\frac{p_{XY}(\mathbf{x},\mathbf{y})}{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}} - 1\Big)\sqrt{\frac{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})}}$, and $D^\diamond(\mathbf{x},\mathbf{y}) = \frac{p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})}$. The supervised implementation of the objective function in (28) is achieved by using Theorem 6.1

$$\mathcal{J}_{HD}(D) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\Big[\mathbb{E}_{\mathbf{y}\sim p_{Y|X}(\mathbf{y}|\mathbf{x})}\Big[-\sqrt{\mathbf{D}(\mathbf{y})}\mathbf{1}_m(\mathbf{x})\Big]\Big] - \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\Big[\frac{1}{\sqrt{\mathbf{D}(\mathbf{y})}}\mathbf{1}_m\Big]. \tag{29}$$

A.4. GAN-Based Objective Functions

GAN: Theorem 3.1 leads to the variational representation of the GAN divergence, when substituting $f^*_{u,GAN}$ listed in Table 2 in (7)

$$\mathcal{J}_{GAN}(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log(1-D(\mathbf{x},\mathbf{y}))\Big] + \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|\log(D(\mathbf{x},\mathbf{y}))\Big], \tag{30}$$

where $T^\diamond(\mathbf{x},\mathbf{y}) = \log\left(\frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})+|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}\right)$ and $D^\diamond(\mathbf{x},\mathbf{y}) = \frac{p_Y(\mathbf{y})}{p_Y(\mathbf{y})+p_{XY}(\mathbf{x},\mathbf{y})}$. The supervised implementation of the objective function in (30) is

$$\mathcal{J}_{GAN}(D) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\Big[\mathbb{E}_{\mathbf{y}\sim p_{Y|X}(\mathbf{y}|\mathbf{x})}\Big[\log(\mathbf{1}_m - \mathbf{D}(\mathbf{y}))\cdot\mathbf{1}_m(\mathbf{x})\Big]\Big] + \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\Big[\log(\mathbf{D}(\mathbf{y}))\mathbf{1}_m\Big]. \tag{31}$$

A.5. P-Based Objective Functions

Pearson χ^2: Theorem 3.1 leads to the variational representation of the Pearson χ^2 divergence, when substituting $f^*_{u,P}$ listed in Table 2 in (7)

$$\mathcal{J}_P(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[2(D(\mathbf{x},\mathbf{y})-1)\Big] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|D^2(\mathbf{x},\mathbf{y})\Big], \tag{32}$$

where $T^\diamond(\mathbf{x},\mathbf{y}) = 2\left(\frac{p_{XY}(\mathbf{x},\mathbf{y})}{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}-1\right)$, and $D^\diamond(\mathbf{x},\mathbf{y}) = \frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_Y(\mathbf{y})}$. The supervised implementation of the objective function in (32) is achieved by using Theorem 6.1

$$\mathcal{J}_P(D) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\Big[\mathbb{E}_{\mathbf{y}\sim p_{Y|X}(\mathbf{y}|\mathbf{x})}\Big[2(\mathbf{D}(\mathbf{y})-\mathbf{1}_m)\mathbf{1}_m(\mathbf{x})\Big]\Big] - \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\Big[\mathbf{D}(\mathbf{y})\mathbf{D}^T(\mathbf{y})\Big], \tag{33}$$

where $\mathbf{D}(\mathbf{y})\mathbf{D}^T(\mathbf{y}) = \sum_{i=1}^m D^2(\mathbf{x}_i,\mathbf{y})$.

A.6. SL-Based Objective Functions

The unsupervised and supervised versions of the objective function corresponding to the shifted log divergence are discussed in this paragraph. For completeness, we report here the generator function and its Fenchel conjugate.
$f_{u,SL}(u) = -|\mathcal{T}_x|\log(u+|\mathcal{T}_x|)$, $f^*_{u,SL}(t) = -|\mathcal{T}_x|(\log(-t)+t)$.

Shifted log: Theorem 4.1 leads to the objective function in (15) when

$$k(p_{X|Y}(\mathbf{x}|\mathbf{y})) = \frac{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})+p_{XY}(\mathbf{x},\mathbf{y})} \tag{34}$$

$$g_1(D,k;1,0) = -\frac{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})+p_{XY}(\mathbf{x},\mathbf{y})}{D(\mathbf{x},\mathbf{y})}. \tag{35}$$

We report here for completeness the unsupervised objective function in (15)

$$\mathcal{J}_{SL}(D) = -\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[D(\mathbf{x},\mathbf{y})\Big] + \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_U(\mathbf{x})p_Y(\mathbf{y})}\Big[|\mathcal{T}_x|\Big(\log(D(\mathbf{x},\mathbf{y}))-D(\mathbf{x},\mathbf{y})\Big)\Big], \tag{36}$$

which can be obtained from Theorem 3.1 with $T^\diamond(\mathbf{x},\mathbf{y}) = -\frac{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})+p_{XY}(\mathbf{x},\mathbf{y})}$ and $D^\diamond(\mathbf{x},\mathbf{y}) = \frac{p_Y(\mathbf{y})}{p_Y(\mathbf{y})+p_{XY}(\mathbf{x},\mathbf{y})}$. The supervised implementation of the objective function in (36) is achieved by using Theorem 6.1

$$\mathcal{J}_{SL}(D) = -\mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\Big[\mathbb{E}_{\mathbf{y}\sim p_{Y|X}(\mathbf{y}|\mathbf{x})}\Big[\mathbf{D}(\mathbf{y})\mathbf{1}_m(\mathbf{x})\Big]\Big] + \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\Big[\Big(\log(\mathbf{D}(\mathbf{y}))-\mathbf{D}(\mathbf{y})\Big)\mathbf{1}_m\Big]. \tag{37}$$

B. Appendix: Proofs

B.1. Proof of Lemma B.1

Lemma B.1. *Let $f : \mathbb{R}_+ \longrightarrow \mathbb{R}$ be the generator function of any f-divergence. Then, it holds:*

$$(f^*)'(t) = (f')^{-1}(t) \tag{38}$$

Proof. Let us recall the definition of Fenchel conjugate, to report a self-contained proof:

$$f^*(t) = \sup_{u \in \mathbb{R}} \{ut - f(u)\}. \tag{39}$$

Then, in order to find $\hat{u}$ that achieves the supremum, we impose

$$\frac{\partial}{\partial u} \{ut - f(u)\} = 0, \tag{40}$$

that implies $f'(u) = t$. The condition (40) can be imposed because $f(\cdot)$ is a convex function. Thus,

$$\hat{u} = (f')^{-1}(t) \tag{41}$$

Then, substituting (41) in the definition of the fenchel conjugate, it becomes:

$$f^*(t) = (f')^{-1}(t)t - f((f')^{-1}(t)). \tag{42}$$

Then, by computing the first derivative w.r.t. t:

$$(f^*)'(t) = ((f')^{-1})'(t)t + (f')^{-1}(t) - \underbrace{f'((f')^{-1}(t))}_{=t}((f')^{-1})'(t). \tag{43}$$

The first and third terms cancel out, leading to (38). $\square$

B.2. Proof of Lemma B.2

Lemma B.2. *Let $f : \mathbb{R}_+ \longrightarrow \mathbb{R}$ be the generator function of any f-divergence, $f^*(\cdot)$ its Fenchel conjugate, and $K > 0$ a constant. Then, $f_u^*(t) \triangleq Kf^*(t)$ is the Fenchel conjugate of a valid f-divergence.*

Proof. Firstly, we must prove the convexity of $f_u(u)$. Since $f^*(t)$ is a convex function, then also $f_u^*(t)$ is a convex function because K is a positive constant (if the second derivative of $f^*(t)$ is non-negative, then multiplying it by a positive constant will result in a non-negative function). If $f_u^*(t)$ is a convex function, then also $f_u(u)$ is convex, by definition, because it is computed as the Fenchel conjugate (i.e., the convex conjugate).

Secondly, if $f_u(1) = C$, then $f_u(1) = 0$ is achieved by subtracting C to the $f_u(u)$ obtained from the computation of the Fenchel conjugate of $f_u^*(t)$. The subtraction of C to $f_u(u)$ has just a translation effect, not affecting the training process. To prove it, let consider $f_u(u) = f_u^\bullet(u) - C$, with $f_u(1) = 0$ (thus $f_u^\bullet(1) = C$). Then,

$$f_u^*(t) = \sup_u [ut - f_u(u)] = \sup_u [ut - f_u^\bullet(u) + C] \tag{44}$$

The u that maximizes $ut - f_u^\bullet(u) + C$ (referred to as $\hat{u}$) is obtained by imposing the first derivative w.r.t. u equal to zero (similarly to (40) and (41) in the proof of Lemma B.1). Therefore, $\hat{u}$ is not influenced by C, since C disappears when computing the first derivative. Then, the value of $\hat{u}$ is substituted in $ut - f_u^\bullet(u) + C$. Therefore, we obtain $f_u^*(t) = \hat{u}t - f_u^\bullet(\hat{u}) + C$, where the first two terms $\hat{u}t$ and $f_u^\bullet(\hat{u})$ do not depend on C. Thus, C becomes a constant in $f_u^*(t)$, which implies that it becomes an additive constant in the objective function (see (7)), which does not have any effect on the training procedure. Finally, $f_u(1) = C$ (thus $f_u^*(t) = \hat{u}t - f_u^\bullet(\hat{u}) + C$) does not impact the estimate of $p_{X|Y}$, since it is computed using $(f_u^*)'(\cdot)$, which is not affected by C. $\square$

B.3. Proof of Theorem 3.1

Theorem 3.1. *Let X and Y be the random vectors with probability density functions $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Assume $\mathbf{y} = H(\mathbf{x})$, where $H(\cdot)$ is a stochastic function, then $p_{XY}(\mathbf{x}, \mathbf{y})$ is the joint density. Define $\mathcal{T}_x$ to be the support of X and $p_U(\mathbf{x})$ to be a uniform distribution with support $\mathcal{T}_x$. Let $f_u : \mathbb{R}_+ \longrightarrow \mathbb{R}$ be a convex function such that $f_u(1) = 0$, and f_u^* be the Fenchel conjugate of f_u. Let $\mathcal{J}_f(T)$ be the objective function defined as*

$$\mathcal{J}_f(T) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim p_{XY}(\mathbf{x}, \mathbf{y})} \left[T(\mathbf{x}, \mathbf{y}) \right] - \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim p_U(\mathbf{x}) p_Y(\mathbf{y})} \left[f_u^*(T(\mathbf{x}, \mathbf{y})) \right]. \tag{45}$$

Then,

$$T^\diamond(\mathbf{x}, \mathbf{y}) = \underset{T \in \mathcal{T}}{\arg\max} \, \mathcal{J}_f(T) \tag{46}$$

leads to the estimation of the posterior density

$$\hat{p}_{X|Y}(\mathbf{x}|\mathbf{y}) = \frac{p_{XY}(\mathbf{x}, \mathbf{y})}{p_Y(\mathbf{y})} = \frac{(f_u^*)'(T^\diamond(\mathbf{x}, \mathbf{y}))}{|\mathcal{T}_x|}, \tag{47}$$

where $T^\diamond(\mathbf{x}, \mathbf{y})$ is parametrized by an artificial neural network.

Proof. From (Nguyen et al., 2010), $T^\diamond(\mathbf{x}, \mathbf{y})$ achieved when maximizing (45) is

$$T^\diamond(\mathbf{x}, \mathbf{y}) = f_u' \left(\frac{p_{XY}(\mathbf{x}, \mathbf{y})}{p_U(\mathbf{x}) p_Y(\mathbf{y})} \right), \tag{48}$$

as defined in (6). Thus, (47) is equivalent to

$$\frac{(f_u^*)'(T^\diamond(\mathbf{x}, \mathbf{y}))}{|\mathcal{T}_x|} = \frac{(f_u')^{-1}(T^\diamond(\mathbf{x}, \mathbf{y}))}{|\mathcal{T}_x|} \tag{49}$$

$$= \frac{(f_u')^{-1} \left(f_u' \left(\frac{p_{XY}(\mathbf{x}, \mathbf{y})}{p_U(\mathbf{x}) p_Y(\mathbf{y})} \right) \right)}{|\mathcal{T}_x|} \tag{50}$$

$$= \frac{p_{XY}(\mathbf{x}, \mathbf{y})}{|\mathcal{T}_x| p_U(\mathbf{x}) p_Y(\mathbf{y})} \tag{51}$$

$$= \frac{p_{XY}(\mathbf{x}, \mathbf{y})}{p_Y(\mathbf{y})}, \tag{52}$$

where the equality in (49) is proved in Lemma B.1 in Appendix B, while (50) is obtained by substituting (48) in (49). Since $p_U(\mathbf{x}) = \frac{1}{|\mathcal{T}_x|}$, as it is the uniform probability density function (pdf) over $\mathcal{T}_x$, the thesis follows. From (Tonello & Letizia, 2022), the usage of the uniform probability density function $p_U(\mathbf{x})$ is fundamental to define the objective function in (45). Its importance derives from the need of the discriminator to be fed with both X and Y realizations.

$\square$

B.4. Proof of Lemma 3.2

Lemma 3.2. *Let the artificial neural network $D(\cdot) \in \mathcal{D}$ be with enough capacity and training time (i.e., in the nonparametric limit). Assume the gradient ascent update rule $D^{(i+1)} = D^{(i)} + \mu \nabla \mathcal{J}_f(D^{(i)})$ converges to*

$$D^\diamond = \underset{D \in \mathcal{D}}{\arg\max} \, \mathcal{J}_f(D), \tag{53}$$

where $\mathcal{J}_f(D)$ is defined as in (7),with the change of variable $D = r^{-1}(T)$. Then, the difference between the optimal posterior probability $p^\diamond$ and its estimate $p^{(i)}$ at iteration i is

$$p^\diamond - p^{(i)} \simeq \frac{1}{|\mathcal{T}_x|} \left(\delta^{(i)} \left[(f_u^*)''(r(D^{(i)})) \right] \right), \tag{54}$$

where $\delta^{(i)} = r(D^\diamond) - r(D^{(i)})$, and $\mu > 0$ the learning rate. If $D^\diamond$ corresponds to the global optimum achieved by using the gradient ascent method, the posterior probability estimator in (9) converges to the real value of the posterior density.

Proof. The proof follows a procedure similar to Lemma 3 in (Letizia et al., 2023a). We define $\delta^{(i)} = r(D^\diamond) - r(D^{(i)})$ as the difference between the optimum $T^\diamond$ and the one achieved at the ith iteration of the training procedure, when using a gradient ascent update method. Define $p^{(i)}$ and $p^\diamond$ as the posterior probability estimate at iteration i and the optimum, respectively. Then,

$$p^\diamond - p^{(i)} = \frac{1}{|\mathcal{T}_x|}\left((f_u^*)'(r(D^\diamond)) - (f_u^*)'(r(D^{(i)}))\right) \tag{55}$$

$$= \frac{1}{|\mathcal{T}_x|}\left((f_u^*)'(r(D^\diamond)) - (f_u^*)'(r(D^\diamond) - \delta^{(i)})\right) \tag{56}$$

$$\simeq \frac{1}{|\mathcal{T}_x|}\left(\delta^{(i)}\left[(f_u^*)''(r(D^\diamond) - \delta^{(i)})\right]\right), \tag{57}$$

where the last step develops from the first order Taylor expansion in $r(D^\diamond) - \delta^{(i)}$. From (57), the thesis in (54) follows. If the gradient ascent method converges towards the maximum, $\delta^{(i)} \longrightarrow 0$. Thus, when $i \longrightarrow \infty$, $|p^\diamond - p^{(i)}| \longrightarrow 0$. $\qquad\square$

B.5. Proof of Theorem 4.1

Theorem 4.1. *Let X and Y be the random vectors with probability density functions $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Assume $\mathbf{y} = H(\mathbf{x})$, where $H(\cdot)$ is a stochastic function, then $p_{XY}(\mathbf{x}, \mathbf{y})$ is the joint density. Let $\mathcal{T}_x$ and $\mathcal{T}_y$ be the support of $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Let the discriminator $D(\mathbf{x}, \mathbf{y})$ be a scalar function of $\mathbf{x}$ and $\mathbf{y}$. Let $k(\cdot)$ be any deterministic and invertible function. Then, the posterior density is estimated as*

$$\hat{p}_{X|Y}(\mathbf{x}|\mathbf{y}) = k^{-1}(D^\diamond(\mathbf{x}, \mathbf{y})), \tag{58}$$

where $D^\diamond(\mathbf{x}, \mathbf{y})$ is the optimal discriminator obtained by maximizing

$$\mathcal{J}(D) = \int_{\mathcal{T}_x} \int_{\mathcal{T}_y} \tilde{\mathcal{J}}(D) d\mathbf{x} d\mathbf{y}, \tag{59}$$

for all concave functions $\tilde{\mathcal{J}}(D)$ such that their first derivative is

$$\frac{\partial \tilde{\mathcal{J}}(D)}{\partial D} = \left(D(\mathbf{x}, \mathbf{y}) - k(p_{X|Y}(\mathbf{x}|\mathbf{y}))\right)g_1(D, k)$$

$$\triangleq g(D, k) \tag{60}$$

with $g_1(D, k) \neq 0$ deterministic, and $\frac{\partial g(D, k)}{\partial D} \leq 0$.

Proof. A necessary condition to maximize $\mathcal{J}(D)$ requires to set the first derivative of the integrand $\tilde{\mathcal{J}}(D)$ w.r.t. D equal to zero. Since $g_1(D, k) \neq 0$, from (60) easily follows

$$D^\diamond(\mathbf{x}, \mathbf{y}) = k(p_{X|Y}(\mathbf{x}|\mathbf{y})) = k\left(\frac{p_{XY}(\mathbf{x}, \mathbf{y})}{p_Y(\mathbf{y})}\right). \tag{61}$$

The concavity of $\tilde{\mathcal{J}}(D)$ is obtained by imposing the first derivative of $g(D, k)$ with respect to D to be nonpositive, i.e.,

$$0 \geq \frac{\partial}{\partial D}\left\{\left(D(\mathbf{x}, \mathbf{y}) - k(p_{X|Y}(\mathbf{x}|\mathbf{y}))\right)g_1(D, k)\right\}$$

$$= g_1(D, k) + \left(D(\mathbf{x}, \mathbf{y}) - k(p_{X|Y}(\mathbf{x}|\mathbf{y}))\right)\frac{\partial g_1(D, k)}{\partial D}. \tag{62}$$

Therefore, the stationary point $D^\diamond(\mathbf{x}, \mathbf{y})$ corresponds to a maximum. $\qquad\square$

B.6. Proof of Corollary 4.2

Corollary 4.2. *Let $\mathcal{J}(D)$ be defined as in Theorem 4.1. Let*

$$k(p_{X|Y}(\mathbf{x}|\mathbf{y})) = k\left(\frac{p_{XY}(\mathbf{x}, \mathbf{y})}{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}\right) = \frac{p_0}{p_1}. \tag{63}$$

Let

$$g_1(D, k; \alpha, \beta) \triangleq -\frac{p_1}{D^\alpha}\left(\frac{1}{1-D}\right)^\beta, \tag{64}$$

where $\alpha, \beta \in \mathbb{Q}$. Then, the objective functions in Table 1 are obtained from Theorem 4.1 by using $k(\cdot)$ and $g_1(\cdot)$ as defined in the set $\mathcal{F}_{k,g_1} \triangleq \{(k(x), (\alpha, \beta))_f\}$:

$$\mathcal{F}_{k,g_1} = \left\{ \left(x, (1,0)\right)_{KL}, \left(-\frac{1}{x}, (1,0)\right)_{RKL}, \left(\frac{1}{x}, \left(\frac{3}{2}, 0\right)\right)_{HD}, \left(\frac{1}{1+x}, (1,1)\right)_{GAN}, \left(x, (0,0)\right)_P \right\} \tag{65}$$

Proof. Let rewrite for completeness

$$k_{KL}(p_{X|Y}(\mathbf{x}|\mathbf{y})) = p_{X|Y}(\mathbf{x}|\mathbf{y}), \tag{66}$$

$$k_{RKL}(p_{X|Y}(\mathbf{x}|\mathbf{y})) = -\frac{1}{p_{X|Y}(\mathbf{x}|\mathbf{y})}, \tag{67}$$

$$k_{HD}(p_{X|Y}(\mathbf{x}|\mathbf{y})) = \frac{1}{p_{X|Y}(\mathbf{x}|\mathbf{y})}, \tag{68}$$

$$k_{GAN}(p_{X|Y}(\mathbf{x}|\mathbf{y})) = \frac{1}{1 + p_{X|Y}(\mathbf{x}|\mathbf{y})}, \tag{69}$$

$$k_P(p_{X|Y}(\mathbf{x}|\mathbf{y})) = p_{X|Y}(\mathbf{x}|\mathbf{y}), \tag{70}$$

and

$$g_{1_{KL}}(D, k) = -\frac{p_U(\mathbf{x})p_Y(\mathbf{y})|\mathcal{T}_x|}{D(\mathbf{x}, \mathbf{y})}, \tag{71}$$

$$g_{1_{RKL}}(D, k) = \frac{p_{XY}(\mathbf{x}, \mathbf{y})}{D(\mathbf{x}, \mathbf{y})}, \tag{72}$$

$$g_{1_{HD}}(D, k) = -\frac{p_{XY}(\mathbf{x}, \mathbf{y})}{D^{\frac{3}{2}}(\mathbf{x}, \mathbf{y})}, \tag{73}$$

$$g_{1_{GAN}}(D, k) = -\frac{p_U(\mathbf{x})p_Y(\mathbf{y})|\mathcal{T}_x| + p_{XY}(\mathbf{x}, \mathbf{y})}{D(\mathbf{x}, \mathbf{y})(1 - D(\mathbf{x}, \mathbf{y}))}, \tag{74}$$

$$g_{1_P}(D, k) = -p_U(\mathbf{x})p_Y(\mathbf{y})|\mathcal{T}_x|, \tag{75}$$

where the dependence of $g_1(\cdot)$ from k is intrinsic in p_1. For instance, the KL and RKL divergences are obtained by using the same values of α and β in (64), but since $k_{KL}(\cdot) \neq k_{RKL}(\cdot)$, $g_{1_{KL}}(\cdot) \neq g_{1_{RKL}}(\cdot)$. $g_1(\cdot) \neq 0$ because by definition each pdf is different from 0 in its support. After a substitution in (14), we obtain

$$g_{KL}(D, k) = \frac{p_{XY}(\mathbf{x}, \mathbf{y})}{D(\mathbf{x}, \mathbf{y})} - p_U(\mathbf{x})p_Y(\mathbf{y})|\mathcal{T}_x|, \tag{76}$$

$$g_{RKL}(D, k) = -p_{XY}(\mathbf{x}, \mathbf{y}) - \frac{p_Y(\mathbf{y})}{D(\mathbf{x}, \mathbf{y})}, \tag{77}$$

$$g_{HD}(D, k) = -\frac{p_{XY}(\mathbf{x}, \mathbf{y})}{\sqrt{D(\mathbf{x}, \mathbf{y})}} + \frac{p_U(\mathbf{x})p_Y(\mathbf{y})}{\sqrt{D^3(\mathbf{x}, \mathbf{y})}}, \tag{78}$$

$$g_{GAN}(D, k) = \frac{p_U(\mathbf{x})p_Y(\mathbf{y})|\mathcal{T}_x|}{D(\mathbf{x}, \mathbf{y})} - \frac{p_{XY}(\mathbf{x}, \mathbf{y})}{1 - D(\mathbf{x}, \mathbf{y})}, \tag{79}$$

$$g_P(D, k) = p_{XY}(\mathbf{x}, \mathbf{y}) - p_U(\mathbf{x})p_Y(\mathbf{y})|\mathcal{T}_x|D(\mathbf{x}, \mathbf{y}). \tag{80}$$

After the integration w.r.t. D and the substitution in (13), the objective functions in Table 1 are attained. $\square$

B.7. Proof of Theorem 5.1

Theorem 5.1. *Let X and Y be two random vectors with pdfs $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$, respectively. Assume $Y = H(X)$, with $H(\cdot)$ stochastic function, then let $p_{XY}(\mathbf{x}, \mathbf{y})$ be the joint density. Let $\mathcal{T}_x$ be the support of X. Let $p_U(\mathbf{x})$ be a uniform pdf*

having the same support $\mathcal{T}_x$. The maximization of the objective function

$$\mathcal{J}_{SL}(D) = -\mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim p_{XY}(\mathbf{x},\mathbf{y})} \Big[D(\mathbf{x},\mathbf{y}) \Big] + \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim p_U(\mathbf{x})p_Y(\mathbf{y})} \Big[|\mathcal{T}_x| \big(\log(D(\mathbf{x},\mathbf{y})) - D(\mathbf{x},\mathbf{y}) \big) \Big], \tag{81}$$

leads to the optimal discriminator output

$$D^{\diamond}(\mathbf{x},\mathbf{y}) = \arg\max_D \mathcal{J}_{SL}(D) = \frac{1}{1 + p_{X|Y}(\mathbf{x},\mathbf{y})}, \tag{82}$$

and the posterior density estimate is computed as

$$\hat{p}_{X|Y}(\mathbf{x}|\mathbf{y}) = \frac{1 - D^{\diamond}(\mathbf{x},\mathbf{y})}{D^{\diamond}(\mathbf{x},\mathbf{y})}. \tag{83}$$

Proof. Following Theorem 4.1, the proof starts by inverting (12). Specifically, we set $k(x) = \frac{1}{1+x}$ and from (82), by expressing the posterior density as the density ratio p_{XY}/p_Y, we achieve

$$D^{\diamond}(\mathbf{x},\mathbf{y}) - \frac{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y}) + p_{XY}(\mathbf{x},\mathbf{y})} = 0, \tag{84}$$

where $p_U(\mathbf{x}) = 1/|\mathcal{T}_x|$. Then, (64) with $\alpha = 1$ and $\beta = 0$ becomes

$$g_1(D, k; 1, 0) = -\frac{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y}) + p_{XY}(\mathbf{x},\mathbf{y})}{D(\mathbf{x},\mathbf{y})}. \tag{85}$$

Then, (85) is substituted in (14), obtaining

$$g(D, k) = \frac{\partial}{\partial D} \tilde{\mathcal{J}}(D) = - \Big(|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y}) + p_{XY}(\mathbf{x},\mathbf{y}) \Big) + \frac{|\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y})}{D(\mathbf{x},\mathbf{y})}. \tag{86}$$

The computation of the integral of (86) with respect to the discriminator's output D leads to

$$\tilde{\mathcal{J}}(D) = -p_{XY}(\mathbf{x},\mathbf{y})D(\mathbf{x},\mathbf{y}) + |\mathcal{T}_x|p_U(\mathbf{x})p_Y(\mathbf{y}) \Big(\log(D(\mathbf{x},\mathbf{y})) - D(\mathbf{x},\mathbf{y}) \Big), \tag{87}$$

which proves the statement of the theorem, since

$$\frac{\partial g(D, k)}{\partial D} = -\frac{p_Y(\mathbf{y})}{D^2(\mathbf{x},\mathbf{y})} \leq 0. \tag{88}$$

Given the optimum discriminator $D^{\diamond}$, the posterior density estimator in (83) is achieved by inverting (82). $\square$

B.8. Proof of Corollary 5.2

Corollary 5.2. *Define the generator function*

$$f_{u,SL}(u) = -|\mathcal{T}_x| \log(u + |\mathcal{T}_x|) + K, \tag{89}$$

where $K = |\mathcal{T}_x| \log(1 + |\mathcal{T}_x|)$ is constant. Then, $\mathcal{J}_{SL}(D)$ in (15) is the variational representation of $D_{f_{u,SL}}(p_{XY}||p_U p_Y)$.

Proof. By comparing (15) to (7), it is immediate to notice the change of variable $D(\mathbf{x},\mathbf{y}) = -T(\mathbf{x},\mathbf{y})$. Then, the expression of the Fenchel conjugate is attained as $f^*_{u,SL}(t) = -|\mathcal{T}_x|(\log(-t) + t)$ from inspection of the expectation over $p_U(\mathbf{x})p_Y(\mathbf{y})$. The generating function $f_{u,SL}(u) = -|\mathcal{T}_x| \log(u + |\mathcal{T}_x|)$ is computed by using the definition of Fenchel conjugate. We add a constant $K = |\mathcal{T}_x| \log(1 + |\mathcal{T}_x|)$ to the generator function to achieve the condition $f_{u,SL}(1) = 0$, which has no impact on the maximization of the objective function in (15). Lastly, the second derivatives of $f_{u,SL}(u)$ and $f^*_{u,SL}(t)$ are nonpositive functions, proving that the generator function and its Fenchel conjugate are convex. $\square$

B.9. Proof of Corollary 5.3

Corollary 5.3. *Let P and Q be two probability distributions. Let $D_{SL}(P||Q)$ be the f-divergence with generator function $f_{SL}(u)$ as in* (19). *Then,*

$$0 \leq D_{SL}(P||Q) \leq \log(2). \tag{90}$$

Proof. Let $f : \mathbb{R}_+ \longrightarrow \mathbb{R}$ be a convex function with $f(1) = 0$, and $f^\circ : \mathbb{R}_+ \longrightarrow \mathbb{R}$ defined as

$$f^\circ(u) \triangleq uf\left(\frac{1}{u}\right), \tag{91}$$

then f° is also convex and such that $f^\circ(1) = 0$. Then, the *Range of Values Theorem* (Vajda, 1972) sets upper and lower bounds on the value of the f-divergence between two distributions P and Q, depending on $f(u)$ and $f^\circ(u)$:

$$f(1) \leq D_f(P||Q) \leq f(0) + f^\circ(0) \quad \forall Q, P. \tag{92}$$

From which the thesis follows. □

B.10. Proof of Corollary 5.4

Corollary 5.4. *Let $\mathcal{J}_{SL}(D)$ be defined as in* (15). *Let $\mathcal{J}_{GAN}(D)$ be defined as in Table* 1. *Let $D_N^\diamond$ be the discriminator output in a neighborhood of $D^\diamond$ where $\mathcal{J}_{SL}(D_N^\diamond)$ and $\mathcal{J}_{GAN}(D_N^\diamond)$ are concave. Then,*

$$\left| \frac{\partial \mathcal{J}_{GAN}(D_N^\diamond)}{\partial D} \right| \geq \left| \frac{\partial \mathcal{J}_{SL}(D_N^\diamond)}{\partial D} \right|. \tag{93}$$

Proof. To prove (93), we just need to prove

$$\left| \frac{\partial \tilde{\mathcal{J}}_{GAN}(D_N^\diamond)}{\partial D} \right| \geq \left| \frac{\partial \tilde{\mathcal{J}}_{SL}(D_N^\diamond)}{\partial D} \right|, \tag{94}$$

(where $\tilde{\mathcal{J}}$ is the integrand function in (13)) since the inequality between the integrands holds when the integrals are computed over the same interval.
Lemma 3.2 guarantees the convergence to the optimal discriminator. Therefore, let $D_N^\diamond = \frac{p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})+p_Y(\mathbf{y})} + \delta$, with δ arbitrarily small, so that $D_N^\diamond$ belongs to the neighborhood of $D^\diamond$. Then,

$$\frac{\partial}{\partial D}\tilde{\mathcal{J}}_{SL}(D)\bigg|_{D_N^\diamond} = -(p_{XY}(\mathbf{x},\mathbf{y}) + p_Y(\mathbf{y})) + \frac{p_Y(\mathbf{y})}{D_N^\diamond(\mathbf{x},\mathbf{y})} \tag{95}$$

$$\frac{\partial}{\partial D}\tilde{\mathcal{J}}_{GAN}(D)\bigg|_{D_N^\diamond} = -\frac{p_{XY}(\mathbf{x},\mathbf{y})}{1 - D_N^\diamond(\mathbf{x},\mathbf{y})} + \frac{p_Y(\mathbf{y})}{D_N^\diamond(\mathbf{x},\mathbf{y})}$$

$$= -\frac{p_{XY}(\mathbf{x},\mathbf{y})(p_{XY}(\mathbf{x},\mathbf{y}) + p_Y(\mathbf{y}))}{p_{XY}(\mathbf{x},\mathbf{y})(1 - \delta) - \delta p_Y(\mathbf{y})}$$

$$+ \frac{p_Y(\mathbf{y})}{D_N^\diamond(\mathbf{x},\mathbf{y})} \tag{96}$$

By substituting $\gamma = \frac{1}{1-\delta}$, (96) becomes

$$\frac{\partial}{\partial D}\tilde{\mathcal{J}}_{GAN}(D)\bigg|_{D_N^\diamond} \approx -\gamma(p_{XY}(\mathbf{x},\mathbf{y}) + p_Y(\mathbf{y})) + \frac{p_Y(\mathbf{y})}{D_N^\diamond(\mathbf{x},\mathbf{y})}, \tag{97}$$

where $0 < \gamma < 1$ if $\delta < 0$, and $\gamma > 1$ if $\delta > 0$. Thus, the comparison between (95) and (97) leads to the inequalities

$$\left.\frac{\partial}{\partial D}\tilde{\mathcal{J}}_{SL}(D)\right|_{D_N^\diamond} < \left.\frac{\partial}{\partial D}\tilde{\mathcal{J}}_{GAN}(D)\right|_{D_N^\diamond} \qquad if \delta < 0 \tag{98}$$

$$\left.\frac{\partial}{\partial D}\tilde{\mathcal{J}}_{SL}(D)\right|_{D_N^\diamond} > \left.\frac{\partial}{\partial D}\tilde{\mathcal{J}}_{GAN}(D)\right|_{D_N^\diamond} \qquad if \delta > 0. \tag{99}$$

Since $D^\diamond$ corresponds to a maximum, i.e., the sign of the left derivative is positive, and the sign of the right derivative is negative, the statement of the corollary is proved. $\qquad\square$

B.11. Proof of Theorem 6.1

Theorem 6.1. *Let $p_X(\mathbf{x})$ and $p_Y(\mathbf{y})$ be pdfs describing the input and output of a stochastic function $H(\cdot)$, respectively. Let $p_X(\mathbf{x}) \triangleq \sum_{i=1}^m P_X(\mathbf{x}_i)\delta(\mathbf{x} - \mathbf{x}_i)$, where $P_X(\cdot)$ is the probability mass function of X. Let $\mathcal{T}_x$ be the support of $p_X(\mathbf{x})$ and $|\mathcal{T}_x|$ its Lebesgue measure. Let $p_U(\mathbf{x})$ be the uniform discrete probability density function over $\mathcal{T}_x$. Let the discriminator be characterized by a supervised architecture. Then, the objective function in (7) becomes*

$$\mathcal{J}(D) = \mathbb{E}_{\mathbf{x}\sim p_X(\mathbf{x})}\left[\mathbb{E}_{\mathbf{y}\sim p_{Y|X}(\mathbf{y}|\mathbf{x})}\left[r(\boldsymbol{D}(\mathbf{y}))\boldsymbol{I}_m(\mathbf{x})\right]\right] - \mathbb{E}_{\mathbf{y}\sim p_Y(\mathbf{y})}\left[\sum_{i=1}^m f^*\left(r(D(\mathbf{x}_i, \mathbf{y}))\right)\right], \tag{100}$$

where $D(\mathbf{x}_i, \mathbf{y})$ is the i-th component of $\boldsymbol{D}(\mathbf{y})$ and $\boldsymbol{T}(\mathbf{y}) = r(\boldsymbol{D}(\mathbf{y}))$.

Proof. Let the alphabet of X be $\mathcal{A}_x = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. The objective function in (7) can be expressed as

$$\mathcal{J}(D) = \int_{\mathcal{T}_y}\int_{\mathcal{T}_x} p_X(\mathbf{x})p_{Y|X}(\mathbf{y}|\mathbf{x})r(D(\mathbf{x}, \mathbf{y})) - p_U(\mathbf{x})p_Y(\mathbf{y})f_u^*\left(r(D(\mathbf{x}, \mathbf{y}))\right)d\mathbf{x}d\mathbf{y}. \tag{101}$$

Then, let $p_U(\mathbf{x}) \triangleq \sum_{i=1}^m P_U(\mathbf{x}_i)\delta(\mathbf{x} - \mathbf{x}_i)$, with $P_U(\mathbf{x}_i) = \frac{1}{|\mathcal{T}_x|}$.

$$\mathcal{J}(D) = \int_{\mathcal{T}_y}\int_{\mathcal{T}_x}\sum_{i=1}^m P_X(\mathbf{x}_i)\delta(\mathbf{x} - \mathbf{x}_i)p_{Y|X}(\mathbf{y}|\mathbf{x})r(D(\mathbf{x}, \mathbf{y})) - \sum_{i=1}^m \delta(\mathbf{x} - \mathbf{x}_i)p_Y(\mathbf{y})f^*\left(r(D(\mathbf{x}, \mathbf{y}))\right)d\mathbf{x}d\mathbf{y}, \tag{102}$$

where $f_u^*\left(r(D(\mathbf{x}, \mathbf{y}))\right) = |\mathcal{T}_x|f^*\left(r(D(\mathbf{x}, \mathbf{y}))\right)$. Then, by using the indicator property of the delta function, the objective function becomes

$$\mathcal{J}(D) = \int_{\mathcal{T}_y}\sum_{i=1}^m P_X(\mathbf{x}_i)p_{Y|X}(\mathbf{y}|\mathbf{x}_i)r(D(\mathbf{x}_i, \mathbf{y}))d\mathbf{y} - \int_{\mathcal{T}_y}p_Y(\mathbf{y})\sum_{i=1}^m f^*\left(r(D(\mathbf{x}_i, \mathbf{y}))\right)d\mathbf{y}, \tag{103}$$

that is equivalent to (100).

$\qquad\square$

B.12. Proof of Closed Form Posterior in the Exponential Case

Closed Form Posterior Exponential Case: Let define the model $Y = X + N$, where $X \sim exp(\lambda)$, $N \sim exp(\lambda)$. Therefore, $Y \sim \Gamma(2, \frac{1}{\lambda})$ (i.e., a Gamma distribution with shape 2 and rate $\frac{1}{\lambda}$). Let define $Y|X \triangleq Z$, then the cumulative density function (CDF) is

$$P[Z \leq z] = P[N + x_c \leq z] = P[N \leq z - x_c]$$
$$= \left(1 - e^{-\lambda(z - x_c)}\right)\mathbb{1}(z - x_c), \tag{104}$$

where X_c is constant. By deriving the CDF w.r.t. z, we compute the likelihood

$$p_{Y|X}(y|x) = \lambda e^{-\lambda(y|x - x_c)}\mathbb{1}(y|x - x_c) = p_N(n). \tag{105}$$

Then, the posterior probability is computed as

$$p_{X|Y}(x|y) = \frac{p_N(n)p_X(x)}{p_Y(y)}$$

$$= \frac{1}{y}\mathbb{1}(x)\mathbb{1}(y). \tag{106}$$

B.13. Proof of Closed Form Posterior in the Gaussian Case

Closed Form Posterior Gaussian Case: Let define the model $Y = X + N$, where $X \sim \mathcal{N}(0, \Sigma_X)$, $N \sim \mathcal{N}(0, \Sigma_N)$. Thus, $Y|X \sim \mathcal{N}(X, \Sigma_N)$, $Y \sim \mathcal{N}(0, \underbrace{\Sigma_N + \Sigma_X}_{\triangleq \Sigma_Y})$.

The posterior probability is

$$p_{X|Y}(x|y) = \frac{p_{Y|X}(y|x)p_X(x)}{p_Y(y)} \tag{107}$$

$$= \frac{\frac{1}{\sqrt{(2\pi)^d|\Sigma_N|}}e^{-\frac{1}{2}(y-x)^T\Sigma_N^{-1}(y-x)}}{\frac{1}{\sqrt{(2\pi)^d|\Sigma_Y|}}e^{-\frac{1}{2}y^T\Sigma_Y^{-1}y}}$$

$$\cdot \frac{1}{\sqrt{(2\pi)^d|\Sigma_X|}}e^{-\frac{1}{2}x^T\Sigma_X^{-1}x} \tag{108}$$

$$= \frac{1}{\sqrt{(2\pi)^d}}\sqrt{\frac{|\Sigma_Y|}{|\Sigma_N||\Sigma_X|}}\exp\left\{-\frac{1}{2}G\right\}, \tag{109}$$

where

$$G = x^T\Sigma_x^{-1}x + y^T\Sigma_N^{-1}y + x^T\Sigma_N^{-1}x - 2x^T\Sigma_N^{-1}y - y^T\Sigma_Y^{-1}y. \tag{110}$$

In the scalar case, where $d = 1$ and $\Sigma_N = \sigma_N^2$, $\Sigma_X = \sigma_X^2$ and $\Sigma_Y = \sigma_X^2 + \sigma_N^2$, we have:

$$G = \frac{x^2}{\sigma_X^2} + \frac{y^2}{\sigma_N^2} + \frac{x^2}{\sigma_N^2} - 2\frac{xy}{\sigma_N^2} - \frac{y^2}{\sigma_y^2} \tag{111}$$

$$= x^2\left(\frac{1}{\sigma_X^2} + \frac{1}{\sigma_N^2}\right) + y^2\left(\frac{1}{\sigma_N^2} - \frac{1}{\sigma_Y^2}\right) - \frac{2xy}{\sigma_N^2} \tag{112}$$

$$= x^2\left(\frac{\sigma_X^2 + \sigma_N^2}{\sigma_X^2\sigma_N^2}\right) + y^2\left(\frac{\sigma_Y^2 - \sigma_N^2}{\sigma_N^2\sigma_Y^2}\right) - \frac{2xy}{\sigma_N^2} \tag{113}$$

$$= \frac{1}{\sigma_N^2}\left[x^2\frac{\sigma_N^2 + \sigma_X^2}{\sigma_X^2} + y^2\frac{\sigma_x^2}{\sigma_X^2 + \sigma_N^2} - 2xy\right] \tag{114}$$

$$= \frac{1}{\sigma_N^2}\left[x^2k^2 + \frac{y^2}{k^2} - 2xy\right] \tag{115}$$

$$= \frac{1}{\sigma_N^2}\left[kx - \frac{y}{k}\right]^2, \tag{116}$$

where $k \triangleq \frac{\sigma_X^2 + \sigma_N^2}{\sigma_X^2} > 1$.

Then, in the scalar case the posterior density expression becomes

$$p_{X|Y}(x|y) = \frac{1}{\sqrt{2\pi}}\sqrt{\frac{\sigma_Y^2}{\sigma_N^2\sigma_X^2}}e^{-\frac{1}{2\sigma_N^2}\left[kx - \frac{y}{k}\right]^2}. \tag{117}$$

C. Appendix: Bottom-up Approach

In this section, we provide additional insights on the bottom-up approach proposed in Theorem 4.1. The idea is to design the objective function by starting from the desired discriminator's output. When designing an objective function using the bottom-up approach, the choice of $k(\cdot)$ is the fundamental starting point of the procedure, which is summarized in the following.

- $k(\cdot)$ is the first DOF and must be chosen as a deterministic and invertible function in the domain $x > 0$. Some examples are $k(x) = x$ and $k(x) = 1/(1+x)$. Then the posterior pdf must be expressed as the density ratio p_{XY}/p_Y.

- $g_1(\cdot)$ is an additional DOF and can be chosen as defined in Corollary 4.2.

- Substitute $k(\cdot)$ and $g_1(\cdot)$ in (14).

- Compute the integral of (14) w.r.t. D, which leads to $\tilde{\mathcal{J}}(D)$.

- The integral in (13) represents the expectation computed over the supports $\mathcal{T}_x$ and $\mathcal{T}_y$.

This exact procedure is used to prove Theorem 5.1.

In the case of the KL-based objective function that we proposed, to ensure that (12) is a density for the case of discrete X, we use the softmax as the last activation function of the discriminator. However, $k(\cdot)$ is not the softmax function because if it were the softmax, then $D^\diamond(\mathbf{x}, \mathbf{y}) = softmax(p_{X|Y}(\mathbf{x}|\mathbf{y}))$, where $p_{X|Y}(\mathbf{x}|\mathbf{y})$ is the density (i.e., the output of the softmax will be $p_{X|Y}(\mathbf{x}|\mathbf{y})$ itself, thus we do not need to apply additional transformations). Therefore, in the case of the KL divergence, $k^{-1}(x) = x$.

D. Appendix: Additional Numerical Results

D.1. Image Datasets

We study the performance of the proposed objective functions for classification (listed in Appendix A) for four image classification datasets. **MNIST** (LeCun et al., 1998) comprises 10 classes with 60,000 images for training and 10,000 images for testing. **Fashion MNIST** has 10 classes with 60,000 images for training and 10,000 images for testing. **CIFAR10** (Krizhevsky et al., 2009) has 10 classes with 50,000 images for training and 10,000 images for testing. **CIFAR100** (Krizhevsky et al., 2009) has 100 classes with 50,000 images for training and 10,000 images for testing. Table 5 shows the classification accuracy of different network architectures trained by using the supervised versions of the objective functions listed in Table 1 and the one presented in Theorem 5.1 (see Appendix A). The accuracies are in the form $XX \pm YY$, where XX and YY represent the mean and standard deviation, respectively, obtained over multiple runs of the code. The tests are run by using various architectures: VGG (Simonyan & Zisserman, 2015), DLA[2] (Yu et al., 2018), ResNet18 (He et al., 2016b), DenseNet (Huang et al., 2017), PreActResNet (He et al., 2016a), MobileNetV2 (Sandler et al., 2018). Interestingly, from Tab. 5, it can be observed that every network architecture suits some objective functions more than others. For instance, the PreActResNet attains the highest accuracy when trained with the GAN-based objective function (even if the difference in performance is just 0.2% different from the second-best one). Differently, the DenseNet performs optimally in CIFAR10 and CIFAR100 when trained with the KL-based objective function, even if the difference in accuracy w.r.t. the SL-based objective function is minimal. The MobileNetV2 obtains the best performance when trained with the SL-based divergence, and the difference with the second-best objective function is around 2%, which is significant. The choice of the architecture often depends on the goal of the classification algorithm. For embedded systems, light architectures are used. Therefore, the MobileNetV2 is an option. In such a case, the SL divergence is the preferred choice for the network's training.

The SL divergence has been proved to have favourable convergence conditions in Corollary 5.4. We numerically compare the convergence speed of the GAN and SL divergences on the CIFAR10 dataset in Figure 5, demonstrating the effectiveness of Corollary 5.4. The accuracy behavior is attained by averaging over multiple runs of the code. In Figure 5, the accuracy over the test dataset is showed, for each training epoch, with a semi-transparent color (blue and orange for the GAN and SL divergences, respectively). The vivid colors represent a moving average over 5 epochs, helping the clarity of the visualization. Quantitatively, we report in Tab. 6 the difference between the accuracy obtained by training the network with the SL and GAN divergences. In detail, each column is identified by a number k and contains, for various discriminator architectures,

---

[2]The implementation of the DLA is a simplified version of the one presented in the original paper

Table 5. Classification accuracy on MNIST (M), Fashion MNIST (FM), CIFAR10 (C10), and CIFAR100 (C100). The PreActResNet is referred to as PAResNet, while the MobileNetV2 is referred to as MobileNet.

DATASET	MODEL	CE	RKL	HD	GAN	P	SL
M	SMALL	**99.08** ± 0.06	96.05 ± 0.25	98.68 ± 0.05	**99.08** ± 0.07	98.89 ± 0.08	99.03 ± 0.04
FM	SMALL	91.64 ± 0.09	82.63 ± 1.78	90.75 ± 0.13	91.63 ± 0.10	89.86 ± 0.67	**91.83** ± 0.02
C10	SMALL	70.13 ± 0.05	63.59 ± 0.34	69.38 ± 0.28	69.98 ± 0.15	59.62 ± 0.45	**70.87** ± 0.26
	VGG	93.69 ± 0.03	84.24 ± 2.21	93.51 ± 0.06	93.75 ± 0.04	84.79 ± 0.21	**93.93** ± 0.08
	DLA	95.04 ± 0.02	90.83 ± 0.10	94.56 ± 0.11	95.04 ± 0.13	91.61 ± 0.21	**95.31** ± 0.09
	RESNET	95.39 ± 0.04	92.88 ± 0.26	95.15 ± 0.08	95.24 ± 0.06	93.78 ± 0.21	**95.43** ± 0.04
	DENSENET	**95.82** ± 0.06	91.52 ± 0.14	94.93 ± 0.07	95.67 ± 0.02	94.34 ± 0.16	95.53 ± 0.13
	PARESNET	94.30 ± 0.15	89.43 ± 0.39	56.44 ± 0.02	**95.17** ± 0.07	86.24 ± 0.08	95.09 ± 0.02
	MOBILENET	92.59 ± 0.13	83.97 ± 0.21	91.95 ± 0.33	92.37 ± 0.14	84.30 ± 0.32	**93.89** ± 0.15
C100	VGG	72.73 ± 0.30	45.80 ± 2.86	73.51 ± 0.03	68.88 ± 0.20	37.19 ± 0.66	**73.61** ± 0.05
	DLA	76.29 ± 0.43	68.86 ± 1.17	78.63 ± 0.14	77.34 ± 0.22	57.97 ± 0.07	**78.65** ± 0.01
	RESNET	**78.29** ± 0.18	70.68 ± 0.44	77.59 ± 0.06	77.43 ± 0.08	61.12 ± 0.23	78.03 ± 0.04
	DENSENET	**80.09** ± 0.02	72.03 ± 0.23	79.91 ± 0.01	79.32 ± 0.34	62.27 ± 0.21	80.03 ± 0.02
	PARESNET	77.13 ± 0.15	59.28 ± 0.12	76.98 ± 0.05	**77.39** ± 0.18	61.64 ± 0.11	77.19 ± 0.25
	MOBILENET	72.61 ± 0.08	53.17 ± 0.35	73.00 ± 0.30	65.66 ± 0.46	46.00 ± 0.37	**74.78** ± 0.23

Table 6. Difference in speed of convergence of the classification accuracy on CIFAR10, between the SL and GAN divergences. Each column is characterized by a number k, which represents the interval $[1, k]$ of epochs over which the average is computed. Each cell contains the difference of accuracy $A_{SL}^k - A_{GAN}^k$.

MODEL	5	10	25	50	75	100	125	150	175	200
RESNET18	4.15	2.81	1.48	1.43	1.16	1.12	0.95	0.86	0.75	0.67
DLA	5.40	4.39	2.52	1.43	1.15	1.11	1.00	0.91	0.81	0.72
VGG	5.93	7.40	5.19	3.42	2.68	2.36	2.07	1.84	1.63	1.44

the quantity $A_{SL}^k - A_{GAN}^k$, where A_f^k is the average accuracy over multiple runs of the code for the selected f-divergence, in the epochs interval $[1, k]$. Each cell in Table 6 contains a positive value, which shows the average faster convergence property of the SL divergence. Furthermore, the difference in speed convergence is more significant for the VGG training than for the ResNet18 and DLA architectures.
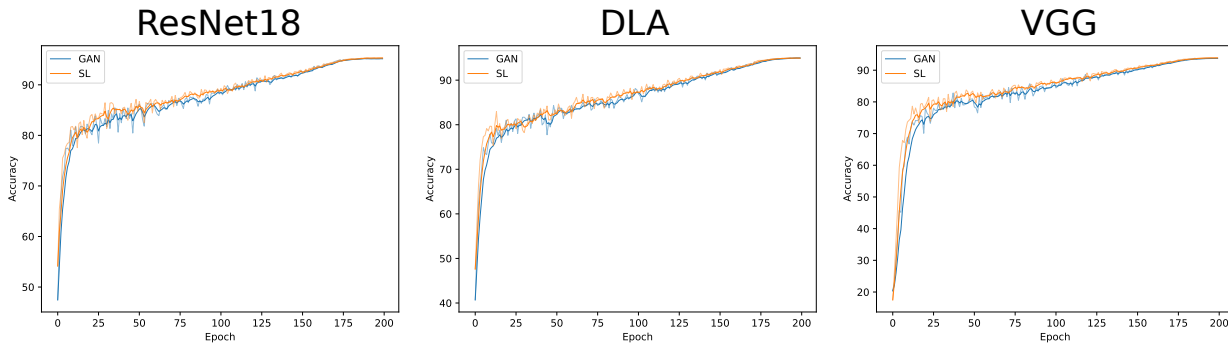

Figure 5. Convergence speed of the test accuracy over 200 training epochs.

The speed of convergence and the upper-boundedness of D_{SL} is (see Corollaries 5.4 and 5.3 in Appendix B.10 and B.9, respectively) confirm the utility of the SL divergence for double optimization problems. For instance, when the objective function of the learning algorithm is formulated as a max-max game. For example, in (Wei & Liu, 2021) the f-mutual information (which is estimated by formulating a maximization problem) is maximized for classification with noisy labels. Similarly, in (Hjelm et al., 2019) the authors maximize the JS-mutual information (where JS refers to the Jensen-Shannon divergence, which is equivalently used in this paper as GAN divergence) for representation learning applications (then the model is extended in (Bachman et al., 2019)). In (Letizia et al., 2023b), the authors formulate a mutual information maximization algorithm to achieve the channel capacity in a data communication system. In (Zhu et al., 2021), the authors

maximize the JS-mutual information for the generation of talking faces.

D.2. Additional Decoding Tasks

D.2.1. AWGN

We analyze the decoding task in presence of additive white Gaussian noise (AWGN) in the communication channel. Let X be a d-dimensional binary vector, and $N \sim \mathcal{N}(0, \Sigma_N)$ be Gaussian noise, with Σ_N diagonal. Let $Y = X + N$ be the output of the communication channel. The SER behavior when varying the SNR is shown in Fig. 6 for each objective function analyzed. To compare the estimated SER, we visualize the SER achieved by the maxL decoder, which corresponds to the optimal decoder for an AWGN channel with uniform distribution $p_X(\mathbf{x})$ (Proakis & Salehi, 2007). The proposed SL divergence achieves the best performance and close to the optimal maxL decoder. In general, different objective functions perform better than $\mathcal{J}_{KL}(D)$.
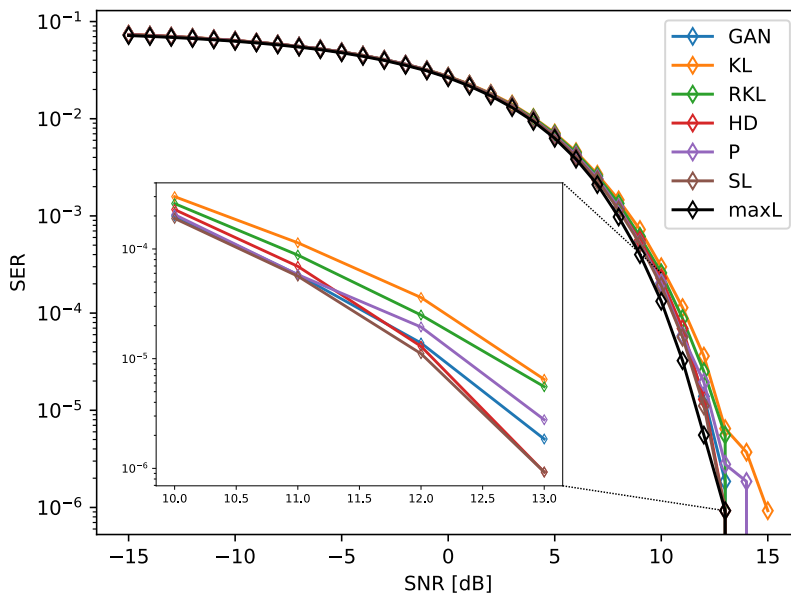

Figure 6. SER achieved in an AWGN channel by the proposed posterior probability estimators.

D.2.2. PAM WITH NON-UNIFORM SOURCE

Similarly to Section 7, we consider a 4-PAM transmission. However, in this case we examine the case where the symbols transmitted do not have a uniform prior probability $p_X(\mathbf{x})$. We define the alphabet of X to be $\mathcal{A}_x = \{x_1, x_2, x_3, x_4\}$ with probabilities $P(x_1) = P(x_2) = P/2$ and $P(x_3) = P(x_4) = (1 - P)/2$, where $P = 0.05$. The SER for various values of SNR is reported in Fig. 7, where the discriminators trained with the supervised versions of the objective functions in Tab. 1 and in (15) (see Appendix A) are compared with the maxL and MAP decoders. Although the maxL decoder is optimal for an AWGN channel, the extreme non-uniformity of the channel significantly impacts its performance. Differently, the optimal MAP decoder knows the distribution of $p_X(\mathbf{x})$. The discriminator trained with the SL and GAN divergences achieves performance close to the optimal MAP decoder.
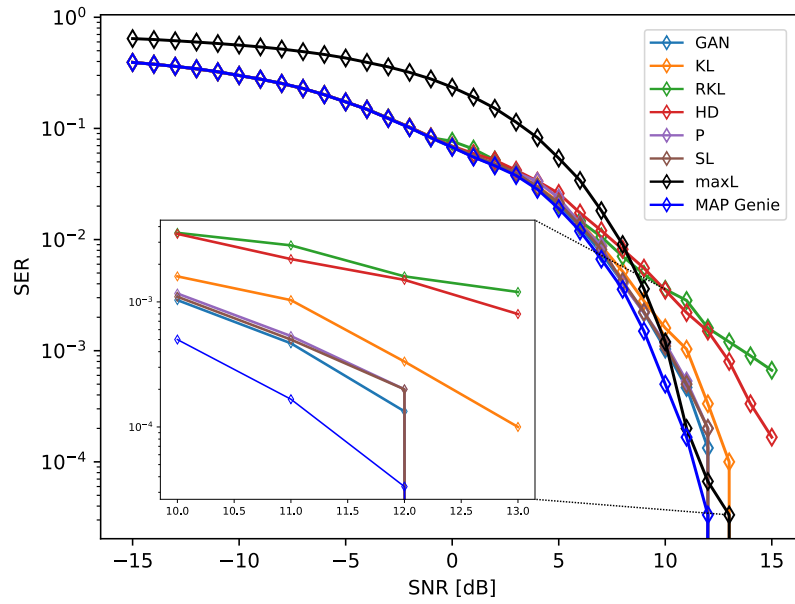
Figure 7. SER achieved in an AWGN channel by a 4-PAM with non-uniform source probability distribution.