A Appendix: DIME Estimators

In this section, we provide a concrete list of DIME estimators obtained using three different f-divergences. In particular, we maximize the value function defined in (5)

$$\mathcal{J}_f(T) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[T\big(\mathbf{x},\mathbf{y}\big) - f^*\Big(T\big(\mathbf{x},\sigma(\mathbf{y})\big)\Big)\Big],$$

over T or its transformation. By doing that, and using (7),

$$I(X;Y) = I_{fDIME}(X;Y) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log\Big(\big(f^*\big)'\big(\hat{T}(\mathbf{x},\mathbf{y})\big)\Big)\Big],$$

we obtain a list of three different MI estimators. The list is used for both commenting on the impact of the f function, referred to as the generator function, and for comparing the estimators discussed in Sec. 2.

We consider the cases when f is the generator of:

a) the KL divergence;

b) the GAN divergence;

c) the Hellinger distance squared.

We report below the derived value functions and the mathematical expressions of the proposed estimators.

A.1 KL divergence

The variational representation of the KL divergence (Nguyen et al., 2010) leads to the NWJ estimator in (30) when $f(u) = u\log(u)$. However, since we are interested in extracting the density ratio, we apply the transformation $T(\mathbf{x}) = \log(D(\mathbf{x}))$. In this way, the lower bound introduced in (5) reads as follows

$$\mathcal{J}_{KL}(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log\big(D\big(\mathbf{x},\mathbf{y}\big)\big)\Big] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}\Big[D\big(\mathbf{x},\mathbf{y}\big)\Big] + 1, \tag{21}$$

which has to be maximized over positive discriminators $D(\cdot)$. As remarked before, we do not use $\mathcal{J}_{KL}$ during the estimation, rather we define the KL-DIME estimator as

$$I_{KL-DIME}(X;Y) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log\Big(\hat{D}(\mathbf{x},\mathbf{y})\Big)\Big], \tag{22}$$

due to the fact that

$$\hat{D}(\mathbf{x},\mathbf{y}) = \arg\max_D \mathcal{J}_{KL}(D) = \frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_X(\mathbf{x})p_Y(\mathbf{y})}. \tag{23}$$

A.2 GAN divergence

Following a similar approach, it is possible to define $f(u) = u\log u - (u+1)\log(u+1) + \log 4$ and $T(\mathbf{x}) = \log(1 - D(\mathbf{x}))$. We derive from Theorem 1 the GAN-DIME estimator obtained via maximization of

$$\mathcal{J}_{GAN}(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log\big(1 - D\big(\mathbf{x},\mathbf{y}\big)\big)\Big] + \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}\Big[\log\big(D\big(\mathbf{x},\mathbf{y}\big)\big)\Big] + \log(4). \tag{24}$$

In fact, at the equilibrium we recover (3), hence

$$I_{GAN-DIME}(X;Y) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log\Big(\frac{1 - \hat{D}(\mathbf{x},\mathbf{y})}{\hat{D}(\mathbf{x},\mathbf{y})}\Big)\Big]. \tag{25}$$

A.3 Hellinger distance

The last example we consider is the generator of the Hellinger distance squared $f(u) = (\sqrt{u}-1)^2$ with the change of variable $T(\mathbf{x}) = 1 - D(\mathbf{x})$. After simple manipulations, we obtain the associated value function as

$$\mathcal{J}_{HD}(D) = 2 - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[D\big(\mathbf{x},\mathbf{y}\big)\Big] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}\left[\frac{1}{D(\mathbf{x},\mathbf{y})}\right], \tag{26}$$

which is maximized for

$$\hat{D}(\mathbf{x},\mathbf{y}) = \arg\max_D \mathcal{J}_{HD}(D) = \sqrt{\frac{p_X(\mathbf{x})p_Y(\mathbf{y})}{p_{XY}(\mathbf{x},\mathbf{y})}}, \tag{27}$$

leading to the HD-DIME estimator

$$I_{HD-DIME}(X;Y) := \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\left[\log\left(\frac{1}{\hat{D}^2(\mathbf{x},\mathbf{y})}\right)\right]. \tag{28}$$

Given that these estimators comprise only one expectation over the joint samples, their variance has different properties compared to the variational ones requiring the partition term such as MINE and NWJ.

B Appendix: Related Work Mutual Information Estimators

In this section, we provide a detailed description of the formulas of the MI estimators we summarized in Section 2.

B.1 MINE

The Donsker-Varadhan dual representation of the KL divergence (Poole et al., 2019; Donsker & Varadhan, 1983) produces an estimate of the MI using the bound optimized by the mutual neural information estimator (MINE) (Belghazi et al., 2018)

$$I_{MINE}(X;Y) = \sup_{\theta\in\Theta} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}[T_\theta(\mathbf{x},\mathbf{y})] - \log(\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}[e^{T_\theta(\mathbf{x},\mathbf{y})}]), \tag{29}$$

where $\theta \in \Theta$ parameterizes a family of functions $T_\theta : \mathcal{X}\times\mathcal{Y} \to \mathbb{R}$ through the use of a deep neural network. However, the logarithm before the expectation in the second term renders MINE a biased estimator. To avoid biased gradients, the authors in (Belghazi et al., 2018) suggested to replace the partition function $\mathbb{E}_{p_X p_Y}[e^{T_\theta}]$ with an exponential moving average over mini-data-batches.

B.2 NWJ

Another variational lower bound is based on the KL divergence dual representation introduced in (Nguyen et al., 2010) (also referred to as f-MINE in (Belghazi et al., 2018))

$$I_{NWJ}(X;Y) = \sup_{\theta\in\Theta} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}[T_\theta(\mathbf{x},\mathbf{y})] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}[e^{T_\theta(\mathbf{x},\mathbf{y})-1}]. \tag{30}$$

Although for a fixed T MINE provides a tighter bound $I_{MINE} \geq I_{NWJ}$, the NWJ estimator is unbiased.

B.3 SMILE

Both MINE and NWJ suffer from high-variance estimations and to combat such a limitation, the SMILE estimator was introduced in (Song & Ermon, 2020). It is defined as

$$I_{SMILE}(X;Y) = \sup_{\theta\in\Theta} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}[T_\theta(\mathbf{x},\mathbf{y})] - \log(\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}[\mathrm{clip}(e^{T_\theta(\mathbf{x},\mathbf{y})}, e^{-\tau}, e^{\tau})]), \tag{31}$$

where $\mathrm{clip}(v,l,u) = \max(\min(v,u),l)$ and it helps to obtain smoother partition functions estimates. SMILE is equivalent to MINE in the limit $\tau \to +\infty$.

B.4 CPC

The MI estimator based on contrastive predictive coding (CPC) (van den Oord et al., 2018) is defined as

$$I_{CPC}(X;Y) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY,N}(\mathbf{x},\mathbf{y})}\left[\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{T_\theta(\mathbf{x}_i,\mathbf{y}_i)}}{\frac{1}{N}\sum_{j=1}^{N}e^{T_\theta(\mathbf{x}_i,\mathbf{y}_j)}}\right)\right], \tag{32}$$

where N is the batch size and $p_{XY,N}$ denotes the joint distribution of N i.i.d. random variables sampled from p_{XY}. CPC provides low variance estimates but it is upper bounded by $\log N$, resulting in a biased estimator.

B.5 NJEE

The neural joint entropy estimator (NJEE) proposed in (Shalev et al., 2022) is based on a classification task. Let X_m be the m-th component of X, with $m \leq d$ and N the batch size. X^k is the vector containing the first k components of X. Let $\hat{H}_N(X_1)$ be the estimated marginal entropy of the first components in X and let $G_{\theta_m}(X_m|X^{m-1})$ be a neural network classifier, where the input is X^{m-1} and the label used is X_m. If $\mathrm{CE}(\cdot)$ is the cross-entropy function, then the MI estimator based on NJEE is defined as

$$I_{NJEE}(X;Y) = \hat{H}_N(X_1) + \sum_{m=2}^{d}\mathrm{CE}(G_{\theta_m}(X_m|X^{m-1})) - \sum_{m=1}^{d}\mathrm{CE}(G_{\theta_m}(X_m|Y,X^{m-1})), \tag{33}$$

where the first two terms of the RHS constitutes the NJEE entropy estimator.

C Appendix: Proofs of Lemmas and Theorems

C.1 Proof of Theorem 1

Theorem 1. *Let $(X,Y) \sim p_{XY}(\mathbf{x},\mathbf{y})$ be a pair of random variables. Let $\sigma(\cdot)$ be a permutation function such that $p_{\sigma(Y)}(\sigma(\mathbf{y})|\mathbf{x}) = p_Y(\mathbf{y})$. Let f^* be the Fenchel conjugate of $f : \mathbb{R}_+ \to \mathbb{R}$, a convex lower semicontinuous function that satisfies $f(1) = 0$ with derivative f'. If $\mathcal{J}_f(T)$ is a value function defined as*

$$\mathcal{J}_f(T) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[T\big(\mathbf{x},\mathbf{y}\big) - f^*\Big(T\big(\mathbf{x},\sigma(\mathbf{y})\big)\Big)\Big], \tag{34}$$

then

$$\hat{T}(\mathbf{x},\mathbf{y}) = \arg\max_T \mathcal{J}_f(T) = f'\left(\frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_X(\mathbf{x})p_Y(\mathbf{y})}\right), \tag{35}$$

and

$$I(X;Y) = I_{fDIME}(X;Y) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\left[\log\Big(\big(f^*\big)'\big(\hat{T}(\mathbf{x},\mathbf{y})\big)\Big)\right]. \tag{36}$$

Proof. From the hypothesis, the value function can be rewritten as

$$\mathcal{J}_f(T) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[T\big(\mathbf{x},\mathbf{y}\big)\Big] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}\Big[f^*\Big(T\big(\mathbf{x},\mathbf{y}\big)\Big)\Big]. \tag{37}$$

The thesis follows immediately from Lemma 1 of (Nguyen et al., 2010). Indeed, the f-divergence D_f can be expressed in terms of its lower bound via Fenchel convex duality

$$D_f(P||Q) \geq \sup_{T\in\mathbb{R}}\Big\{\mathbb{E}_{x\sim p(x)}\big[T(x)\big] - \mathbb{E}_{x\sim q(x)}\big[f^*\big(T(x)\big)\big]\Big\}, \tag{38}$$

where $T : \mathcal{X} \to \mathbb{R}$ and f^* is the Fenchel conjugate of f defined as

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

Therein, it was shown that the bound in (38) is tight for optimal values of $T(x)$ and it takes the following form

$$\hat{T}(x) = f'\bigg(\frac{p(x)}{q(x)}\bigg), \tag{40}$$

where f' is the derivative of f.

The mutual information $I(X;Y)$ admits the KL divergence representation

$$I(X;Y) = D_{KL}(p_{XY}||p_X p_Y), \tag{41}$$

and since the inverse of the derivative of f is the derivative of the conjugate f^*, the density ratio can be rewritten in terms of the optimum discriminator $\hat{T}$

$$\big(f'\big)^{-1}\big(\hat{T}(\mathbf{x},\mathbf{y})\big) = \big(f^*\big)'\big(\hat{T}(\mathbf{x},\mathbf{y})\big) = \frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_X(\mathbf{x})p_Y(\mathbf{y})}. \tag{42}$$

f-DIME finally reads as follows

$$I_{fDIME}(X;Y) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\bigg[\log\bigg(\big(f^*\big)'\big(\hat{T}(\mathbf{x},\mathbf{y})\big)\bigg)\bigg]. \tag{43}$$

□

C.2 Proof of Lemma 1

Lemma 1. *Let the discriminator $T(\cdot)$ be with enough capacity, i.e., in the non parametric limit. Consider the problem*

$$\hat{T} = \arg\max_{T} \mathcal{J}_f(T) \tag{44}$$

where

$$\mathcal{J}_f(T) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[T\big(\mathbf{x},\mathbf{y}\big)\Big] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}\bigg[f^*\bigg(T\big(\mathbf{x},\mathbf{y}\big)\bigg)\bigg], \tag{45}$$

and the update rule based on the gradient descent method

$$T^{(n+1)} = T^{(n)} + \mu\nabla\mathcal{J}_f(T^{(n)}). \tag{46}$$

If the gradient descent method converges to the global optimum $\hat{T}$, the mutual information estimator

$$I(X;Y) = I_{fDIME}(X;Y) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\bigg[\log\bigg(\big(f^*\big)'\big(\hat{T}(\mathbf{x},\mathbf{y})\big)\bigg)\bigg]. \tag{47}$$

converges to the real value of the mutual information $I(X;Y)$.

Proof. For convenience of notation, let the instantaneous mutual information be the random variable defined as

$$i(X;Y) := \log\bigg(\frac{p_{XY}(\mathbf{x},\mathbf{y})}{p_X(\mathbf{x})p_Y(\mathbf{y})}\bigg). \tag{48}$$

It is straightforward to notice that the MI corresponds to the expected value of $i(X;Y)$ over the joint distribution p_{XY}. The solution to (44) is given by (6) of Theorem 1. Let $\delta^{(n)} = \hat{T} - T^{(n)}$ be the displacement between the optimum discriminator $\hat{T}$ and the obtained one $T^{(n)}$ at the iteration n, then

$$\hat{i}_{n,fDIME}(X;Y) = \log\bigg(\big(f^*\big)'\big(T^{(n)}(\mathbf{x},\mathbf{y})\big)\bigg) = \log\bigg(R^{(n)}(\mathbf{x},\mathbf{y})\bigg), \tag{49}$$

where $R^{(n)}(\mathbf{x},\mathbf{y})$ represents the estimated density ratio at the n-th iteration and it is related with the optimum ratio $\hat{R}(\mathbf{x},\mathbf{y})$ as follows

$$\begin{aligned}
\hat{R} - R^{(n)} &= \big(f^*\big)'\big(\hat{T}\big) - \big(f^*\big)'\big(T^{(n)}\big) \\
&= \big(f^*\big)'\big(\hat{T}\big) - \big(f^*\big)'\big(\hat{T} - \delta^{(n)}\big) \\
&\simeq \delta^{(n)} \cdot \Big[\big(f^*\big)''\big(\hat{T} - \delta^{(n)}\big)\Big],
\end{aligned} \tag{50}$$

where the last step follows from a first order Taylor expansion in $\hat{T} - \delta^{(n)}$. Therefore,

$$\begin{aligned}\hat{i}_{n,fDIME}(X;Y) &= \log\big(R^{(n)}\big) \\ &= \log\left(\big(\hat{R}\big)\left(1 - \delta^{(n)} \cdot \frac{\big(f^*\big)''\big(\hat{T} - \delta^{(n)}\big)}{\big(f^*\big)'\big(\hat{T}\big)}\right)\right) \\ &= i(X;Y) + \log\left(1 - \delta^{(n)} \cdot \frac{\big(f^*\big)''\big(\hat{T} - \delta^{(n)}\big)}{\big(f^*\big)'\big(\hat{T}\big)}\right). \end{aligned} \tag{51}$$

If the gradient descent method converges towards the optimum solution $\hat{T}$, $\delta^{(n)} \to 0$ and

$$\begin{aligned}\hat{i}_{n,fDIME}(X;Y) &\simeq i(X;Y) - \delta^{(n)} \cdot \left[\frac{\big(f^*\big)''\big(\hat{T} - \delta^{(n)}\big)}{\big(f^*\big)'\big(\hat{T}\big)}\right] \\ &\simeq i(X;Y) - \delta^{(n)} \cdot \left[\frac{\big(f^*\big)''\big(\hat{T}\big)}{\big(f^*\big)'\big(\hat{T}\big)}\right] \\ &= i(X;Y) - \delta^{(n)} \cdot \left[\frac{\mathrm{d}}{\mathrm{d}T}\log\big(\big(f^*\big)'(T)\big)\Big|_{T=\hat{T}}\right], \end{aligned} \tag{52}$$

where the RHS is itself a first order Taylor expansion of the instantaneous mutual information in $\hat{T}$. In the asymptotic limit ($n \to +\infty$), it holds also for the expected values that

$$|I(X;Y) - \hat{I}_{n,fDIME}(X;Y)| \to 0. \tag{53}$$

□

C.3 Proof of Lemma 2

Lemma 2. *Let $\hat{R} = p_{XY}(\mathbf{x},\mathbf{y})/(p_X(\mathbf{x})p_Y(\mathbf{y}))$ and assume $Var_{p_{XY}}[\log \hat{R}]$ exists. Let p^M_{XY} be the empirical distribution of M i.i.d. samples from p_{XY} and let $\mathbb{E}_{p^M_{XY}}$ denote the sample average over p^M_{XY}. Then, under the randomness of the sampling procedure, it holds that*

$$Var_{p_{XY}}\left[\mathbb{E}_{p^M_{XY}}[\log \hat{R}]\right] \leq \frac{4H^2(p_{XY}, p_X p_Y)\big|\big|\hat{R}\big|\big|_\infty - I^2(X;Y)}{M} \tag{54}$$

where H^2 is the Hellinger distance squared defined as

$$H^2(p,q) = \int_{\mathbf{x}}\left(\sqrt{p(\mathbf{x})} - \sqrt{q(\mathbf{x})}\right)^2 \mathrm{d}\mathbf{x}, \tag{55}$$

and the infinity norm is defined as $||f(x)||_\infty := \sup_{x\in\mathbb{R}} |f(x)|$.

Proof. Consider the variance of $\hat{R}(\mathbf{x},\mathbf{y})$ when $(\mathbf{x},\mathbf{y}) \sim p_{XY}(\mathbf{x},\mathbf{y})$, then

$$\mathrm{Var}_{p_{XY}}[\log \hat{R}] = \mathbb{E}_{p_{XY}}\left[\left(\log \frac{p_{XY}}{p_X p_Y}\right)^2\right] - \left(\mathbb{E}_{p_{XY}}\left[\log \frac{p_{XY}}{p_X p_Y}\right]\right)^2. \tag{56}$$

The power of the log-density ratio is upper bounded as follows (see the approach of Lemma 8.3 in (Ghosal et al., 2000))

$$\mathbb{E}_{p_{XY}}\left[\left(\log \frac{p_{XY}}{p_X p_Y}\right)^2\right] \leq 4H^2(p_{XY}, p_X p_Y)\left|\left|\frac{p_{XY}}{p_X p_Y}\right|\right|_\infty, \tag{57}$$

while the mean squared is the ground-truth mutual information squared, thus

$$\mathrm{Var}_{p_{XY}}[\log \hat{R}] \leq 4H^2(p_{XY}, p_X p_Y)\left|\left|\frac{p_{XY}}{p_X p_Y}\right|\right|_\infty - I^2(X;Y). \tag{58}$$

Finally, the variance of the mean of M i.i.d. random variables yields the thesis

$$\mathrm{Var}_{p_{XY}}\left[\mathbb{E}_{p^M_{XY}}[\log \hat{R}]\right] = \frac{\mathrm{Var}_{p_{XY}}[\log \hat{R}]}{M} \leq \frac{4H^2(p_{XY}, p_X p_Y)\left|\left|\frac{p_{XY}}{p_X p_Y}\right|\right|_\infty - I^2(X;Y)}{M}. \tag{59}$$

□

C.4 Proof of Lemma 3

Lemma 3. *Let $(\mathbf{x}_i, \mathbf{y}_i)$, $\forall i \in \{1, \dots, N\}$, be N data points. Let $\mathcal{J}_f(T)$ be the value function in (5). Let $\mathcal{J}_f^\pi(T)$ and $\mathcal{J}_f^\sigma(T)$ be numerical implementations of $\mathcal{J}_f(T)$ using a random permutation and a random derangement of $\mathbf{y}$, respectively. Denote with K the number of points $\mathbf{y}_k$, with $k \in \{1, \dots, N\}$, in the same position after the permutation (i.e., the fixed points). Then*

$$\mathcal{J}_f^\pi(T) \leq \frac{N-K}{N} \mathcal{J}_f^\sigma(T). \tag{60}$$

Proof. Define $\mathcal{J}_f^\pi(T)$ as the Monte Carlo implementation of $\mathcal{J}_f(T)$ when using the permutation function $\pi(\cdot)$

$$\mathcal{J}_f^\pi(T) = \frac{1}{N} \sum_{i=1}^{N} T(\mathbf{x}_i, \mathbf{y}_i) - \frac{1}{N} \sum_{i=1}^{N} f^*\big(T(\mathbf{x}_i, \mathbf{y}_j)\big), \tag{61}$$

where the pair $(\mathbf{x}_i, \mathbf{y}_j)$ is obtained via a random permutation of the elements of $\mathbf{y}$ as $j = \pi(i)$, $\forall i \in \{1, \dots, N\}$. Since K is a non-negative integer representing the number of fixed points $i = \pi(i)$, the value function can be rewritten as

$$\mathcal{J}_f^\pi(T) = \frac{1}{N} \sum_{i=1}^{N} T(\mathbf{x}_i, \mathbf{y}_i) - \frac{1}{N} \sum_{i=1}^{K} f^*\big(T(\mathbf{x}_i, \mathbf{y}_i)\big) - \frac{1}{N} \sum_{i=1}^{N-K} f^*\big(T(\mathbf{x}_i, \mathbf{y}_{j \neq i})\big), \tag{62}$$

which can also be expressed as

$$\mathcal{J}_f^\pi(T) = \frac{1}{N} \sum_{i=1}^{K} T(\mathbf{x}_i, \mathbf{y}_i) + \frac{1}{N} \sum_{i=1}^{N-K} T(\mathbf{x}_i, \mathbf{y}_i) - \frac{1}{N} \sum_{i=1}^{K} f^*\big(T(\mathbf{x}_i, \mathbf{y}_i)\big) - \frac{1}{N} \sum_{i=1}^{N-K} f^*\big(T(\mathbf{x}_i, \mathbf{y}_{j \neq i})\big). \tag{63}$$

In (63) it is possible to recognize that the second and last term of the RHS constitutes the numerical implementation of $\mathcal{J}_f(T)$ using a derangement strategy on $N - K$ elements, so that

$$\mathcal{J}_f^\pi(T) = \frac{1}{N} \sum_{i=1}^{K} T(\mathbf{x}_i, \mathbf{y}_i) - \frac{1}{N} \sum_{i=1}^{K} f^*\big(T(\mathbf{x}_i, \mathbf{y}_i)\big) + \frac{N-K}{N} J_f^\sigma(T). \tag{64}$$

However, by definition of Fenchel conjugate

$$\frac{1}{N} \sum_{i=1}^{K} T(\mathbf{x}_i, \mathbf{y}_i) - f^*\big(T(\mathbf{x}_i, \mathbf{y}_i)\big) \leq 0, \tag{65}$$

since for $t = 1$

$$u - f^*(u) \leq u - (ut - f(t)) = f(1) = 0. \tag{66}$$

Hence, we can conclude that

$$\mathcal{J}_f^\pi(T) \leq \frac{N-K}{N} J_f^\sigma(T). \tag{67}$$

□

C.5 Proof of Lemma 4

Lemma 4. *Let $\hat{R}$ be the optimal density ratio and let $X \sim \mathcal{N}(0, \sigma_X^2)$ and $N \sim \mathcal{N}(0, \sigma_N^2)$ be uncorrelated scalar Gaussian random variables such that $Y = X + N$. Assume $Var_{p_{XY}}[\log \hat{R}]$ exists. Let p_{XY}^M be the empirical distribution of M i.i.d. samples from p_{XY} and let $\mathbb{E}_{p_{XY}^M}$ denote the sample average over p_{XY}^M. Then, under the randomness of the sampling procedure, it holds that*

$$Var_{p_{XY}}\left[\mathbb{E}_{p_{XY}^M}[\log \hat{R}]\right] = \frac{1 - e^{-2I(X;Y)}}{M}. \tag{68}$$

Proof. From the hypothesis, the density ratio can be rewritten as $\hat{R} = p_N(y-x)/p_Y(y)$ and the output variance is clearly equal to $\sigma_Y^2 = \sigma_X^2 + \sigma_N^2$. Notice that this is equivalent of having correlated random variables X and Y with correlation coefficient ρ, since it is enough to study the case $\sigma_X = \rho$ and $\sigma_N = \sqrt{1-\rho^2}$.

It is easy to verify via simple calculations that

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{p_{XY}}[\log \hat{R}] \\ &= \log \frac{\sigma_Y}{\sigma_N} + \mathbb{E}_{p_{XY}}\left[\frac{y^2}{2\sigma_Y^2} - \frac{(y-x)^2}{2\sigma_N^2}\right] \\ &= \cdots = \log \frac{\sigma_Y}{\sigma_N} = \frac{1}{2}\log\left(1 + \frac{\sigma_X^2}{\sigma_N^2}\right) = -\frac{1}{2}\log\big(1-\rho^2\big). \end{aligned} \tag{69}$$

Similarly,

$$\begin{aligned} &\mathrm{Var}_{p_{XY}}\big[\log \hat{R}\big] = \mathbb{E}_{p_{XY}}\left[\left(\log\left(\frac{\sigma_Y}{\sigma_N}\right) + \frac{y^2}{2\sigma_Y^2} - \frac{(y-x)^2}{2\sigma_N^2}\right)^2\right] - I^2(X;Y) \\ &= \frac{1}{4}\mathbb{E}_{p_{XY}}\left[\left(\frac{y-x}{\sigma_N}\right)^4 + \left(\frac{y}{\sigma_Y}\right)^4 - 2\left(\frac{y}{\sigma_Y}\right)^2\left(\frac{y-x}{\sigma_N}\right)^2\right] \\ &= \cdots = \mathrm{Kurt}[Z]\left(\frac{1}{2} - \frac{\sigma_N^2}{2\sigma_Y^2}\right) - \frac{\sigma_X^2}{2\sigma_Y^2} \\ &= \frac{\sigma_X^2}{\sigma_Y^2} = 1 - \frac{\sigma_N^2}{\sigma_Y^2} = 1 - e^{-2I(X;Y)} = \rho^2, \end{aligned} \tag{70}$$

where the last steps use the fact that the Kurtosis of a normal distribution is 3 and that the mutual information can be expressed as in (69). Finally, the variance of the mean of M i.i.d. random variables yields the thesis

$$\mathrm{Var}_{p_{XY}}\big[\mathbb{E}_{p_{XY}^M}[\log \hat{R}]\big] = \frac{\mathrm{Var}_{p_{XY}}[\log \hat{R}]}{M}. \tag{71}$$

If X and N are multivariate Gaussians with diagonal covariance matrices $\rho^2 \mathbb{I}_{d\times d}$ and $(1-\rho^2)\mathbb{I}_{d\times d}$, the results for both the MI and variance in the scalar case are simply multiplied by d. □

C.6 Proof of Theorem 2

Theorem 2. *Let the discriminator $D(\cdot)$ be with enough capacity. Let N be the batch size and f be the generator of the KL divergence. Let $\mathcal{J}_{KL}^{\pi}(D)$ be defined as*

$$\mathcal{J}_{KL}^{\pi}(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\left[\log\Big(D\big(\mathbf{x},\mathbf{y}\big)\Big) - f^*\left(\log\Big(D\big(\mathbf{x},\pi(\mathbf{y})\big)\Big)\right)\right]. \tag{72}$$

Denote with K the number of indices in the same position after the permutation (i.e., the fixed points), and with $R(\mathbf{x},\mathbf{y})$ the density ratio in (2). Then,

$$\hat{D}(\mathbf{x},\mathbf{y}) = \arg\max_D \mathcal{J}_{KL}^{\pi}(D) = \frac{NR(\mathbf{x},\mathbf{y})}{KR(\mathbf{x},\mathbf{y}) + N - K}. \tag{73}$$

Proof. The idea of the proof is to express $\mathcal{J}_{KL}^{\pi}(D)$ via Monte Carlo approximation, in order to rearrange fixed points, and then go back to Lebesgue integration. The value function $\mathcal{J}_{KL}(D)$ can be written as

$$\mathcal{J}_{KL}(D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_{XY}(\mathbf{x},\mathbf{y})}\Big[\log\big(D(\mathbf{x},\mathbf{y})\big)\Big] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p_X(\mathbf{x})p_Y(\mathbf{y})}\Big[D\big(\mathbf{x},\mathbf{y}\big)\Big] + 1. \tag{74}$$

Similarly to (62), we can express $\mathcal{J}_{KL}^{\pi}(D)$ as

$$\mathcal{J}_{KL}^{\pi}(D) = \frac{1}{N}\sum_{i=1}^{N}\log\big(D(\mathbf{x}_i,\mathbf{y}_i)\big) - \frac{1}{N}\sum_{i=1}^{K} D(\mathbf{x}_i,\mathbf{y}_i) - \frac{1}{N}\sum_{i=1}^{N-K} D(\mathbf{x}_i,\mathbf{y}_{j\neq i}) + 1, \tag{75}$$

where K is the number of fixed points of the permutation $j = \pi(i), \forall i \in \{1, \dots, N\}$. However, when $N \to \infty$, we can use Lebesgue integration and rewrite (75) as

$$\mathcal{J}^{\pi}_{KL}(D) = \int_{\mathbf{x}} \int_{\mathbf{y}} \left(p_{XY}(\mathbf{x}, \mathbf{y}) \log\big(D(\mathbf{x}, \mathbf{y})\big) - \frac{K}{N} p_{XY}(\mathbf{x}, \mathbf{y}) D(\mathbf{x}, \mathbf{y}) \right) \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{y} - \int_{\mathbf{x}} \int_{\mathbf{y}} \frac{N-K}{N} p_X(\mathbf{x}) p_Y(\mathbf{y}) D(\mathbf{x}, \mathbf{y})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{y} + 1. \tag{76}$$

To maximize $\mathcal{J}^{\pi}_{KL}(D)$, it is enough to take the derivative of the integrand with respect to D and equate it to 0, yielding the following equation in D

$$\frac{p_{XY}(\mathbf{x}, \mathbf{y})}{D(\mathbf{x}, \mathbf{y})} - \frac{K}{N} p_{XY}(\mathbf{x}, \mathbf{y}) - \frac{N-K}{N} p_X(\mathbf{x}) p_Y(\mathbf{y}) = 0. \tag{77}$$

Solving for D leads to the thesis

$$\hat{D}(\mathbf{x}, \mathbf{y}) = \frac{NR(\mathbf{x}, \mathbf{y})}{KR(\mathbf{x}, \mathbf{y}) + N - K}, \tag{78}$$

since $\mathcal{J}^{\pi}_{KL}(\hat{D})$ is a maximum being the second derivative w.r.t. D a non-positive function. □

C.7 Proof of Corollary 2.1

Corollary 2.1 (Permutation bound)**.** *Let KL-DIME be the estimator obtained via iterative optimization of $\mathcal{J}^{\pi}_{KL}(D)$, using a batch of size N every training step. Then,*

$$I^{\pi}_{KL-DIME} := \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim p_{XY}(\mathbf{x},\mathbf{y})} \left[\log\left(\hat{D}(\mathbf{x}, \mathbf{y}) \right) \right] < \log(N). \tag{79}$$

Proof. Theorem 2 implies that, when the batch size is much larger than the density ratio ($N >> R$), then the discriminator's output converges to the density ratio. Indeed,

$$\lim_{N \to \infty} \hat{D}(\mathbf{x}, \mathbf{y}) = \lim_{N \to \infty} \frac{NR(\mathbf{x}, \mathbf{y})}{KR(\mathbf{x}, \mathbf{y}) + N - K} = R(\mathbf{x}, \mathbf{y}). \tag{80}$$

Instead, when the density ratio is much larger than the batch size ($R >> N$), then the discriminator's output converges to a constant, in particular

$$\lim_{R \to \infty} \hat{D}(\mathbf{x}, \mathbf{y}) = \lim_{R \to \infty} \frac{NR(\mathbf{x}, \mathbf{y})}{KR(\mathbf{x}, \mathbf{y}) + N - K} = \frac{N}{K}. \tag{81}$$

However, from Lemma 5, it is true that $K = 1$ on average. Therefore, an iterative optimization algorithm leads to an upper-bounded discriminator, since

$$\hat{D}(\mathbf{x}, \mathbf{y}) < N, \tag{82}$$

which implies the thesis. □

C.8 Proof of Lemma 5

Lemma 5 (see (Alon & Spencer, 2016))**.** *The average number of fixed points in a random permutation $\pi(\cdot)$ is equal to 1.*

Proof. Let $\pi(\cdot)$ be a random permutation on $\{1, \dots, N\}$. Let the random variable X represent the number of fixed points (i.e., the number of cycles of length 1) of $\pi(\cdot)$. We define $X = X_1 + X_2 + \cdots + X_N$, where $X_i = 1$ when $\pi(i) = i$, and 0 otherwise. $\mathbb{E}[X]$ is computed by exploiting the linearity property of expectation. Trivially,

$$\mathbb{E}[X_i] = \mathbb{P}[\pi(i) = i] = \frac{1}{N}, \tag{83}$$

which implies

$$\mathbb{E}[X] = \sum_{i=1}^{N} \frac{1}{N} = 1. \tag{84}$$

□

D Appendix: Experimental Details

D.1 Multivariate Linear and Nonlinear Gaussians Experiments

The neural network architectures implemented for the linear and cubic Gaussian experiments are: *joint*, *separable*, *deranged*, and the architecture of NJEE, referred to as *ad hoc*.

Joint architecture. The *joint* architecture is a feed-forward fully connected neural network with an input size equal to twice the dimension of the samples distribution ($2d$), one output neuron, and two hidden layers of 256 neurons each. The activation function utilized in each layer (except from the last one) is ReLU. The number of realizations $(\mathbf{x}, \mathbf{y})$ fed as input of the neural network for each training iteration is N^2, obtained as all the combinations of the samples $\mathbf{x}$ and $\mathbf{y}$ drawn from $p_{XY}(\mathbf{x}, \mathbf{y})$.

Separable architecture. The *separable* architecture comprises two feed-forward neural networks, each one with an input size equal to d, output layer containing 32 neurons and 2 hidden layers with 256 neurons each. The ReLU activation function is used in each layer (except from the last one). The first network is fed in with N realizations of X, while the second one with N realizations of Y.

Deranged architecture. The *deranged* architecture is a feed-forward fully connected neural network with an input size equal to twice the dimension of the samples distribution ($2d$), one output neuron, and two hidden layers of 256 neurons each. The activation function utilized in each layer (except from the last one) is ReLU. The number of realizations $(\mathbf{x}, \mathbf{y})$ the neural network is fed with is $2N$ for each training iteration: N realizations drawn from $p_{XY}(\mathbf{x}, \mathbf{y})$ and N realizations drawn from $p_X(\mathbf{x})p_Y(\mathbf{y})$ using the derangement procedure described in Sec. 5.
The architecture *deranged* is not used for I_{CPC} because in (32) the summation at the denominator of the argument of the logarithm would require neural network predictions corresponding to the inputs $(\mathbf{x}_i, \mathbf{y}_j)$, $\forall i, j \in \{1, \dots, N\}$ with $i \neq j$.

Ad hoc architecture. The *NJEE* MI estimator comprises $2d - 1$ feed-forward neural networks. Each neural network is composed by an input layer with size between 1 and $2d-1$, an output layer containing $N - k$ neurons, with $k \in \mathbb{N}$ small, and 2 hidden layers with 256 neurons each. The ReLU activation function is used in each layer (except from the last one). We implemented a Pytorch (Paszke et al., 2016) version of the code produced by the authors of (Shalev et al., 2022) [2], to unify NJEE with all the other MI estimators.

Each neural estimator is trained using Adam optimizer (Kingma & Ba, 2014), with learning rate 5×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The batch size is initially set to $N = 64$.
For the *Gaussian* setting, we sample a 20-dimensional Gaussian distribution to obtain $\mathbf{x}$ and $\mathbf{n}$ samples, independently. Then, we compute $\mathbf{y}$ as linear combination of $\mathbf{x}$ and $\mathbf{n}$: $\mathbf{y} = \rho\,\mathbf{x} + \sqrt{1-\rho^2}\,\mathbf{n}$, where ρ is the correlation coefficient. For the *cubic* setting, the nonlinear transformation $\mathbf{y} \mapsto \mathbf{y}^3$ is applied to the Gaussian samples. During the training procedure, every $4k$ iterations, the target value of the MI is increased by 2 *nats*, for 5 times, obtaining a target staircase with 5 steps. The change in target MI is obtained by increasing ρ, that affects the true MI according to

$$I(X;Y) = -\frac{d}{2}\log(1-\rho^2). \tag{85}$$

D.1.1 Supplementary Analysis of the MI Estimators Performance

Additional plots reporting the MI estimates obtained from MINE, NWJ, and SMILE with $\tau = \infty$, are outlined in Fig. 6. The variance attained by these algorithms exponentially increases as the true MI grows, as stated in (11).

We report in Fig. 7 the behavior we obtained for I_{SMILE} when the training of the neural network is performed by using the cost function in (31). The training diverges during the first steps when $\tau = 1$ and $\tau = 5$. Differently, when $\tau = \infty$, I_{SMILE} corresponds to I_{MINE} (without the moving average improvement), therefore the MI estimate does not diverge.

[2] https://github.com/YuvalShalev/NJEE

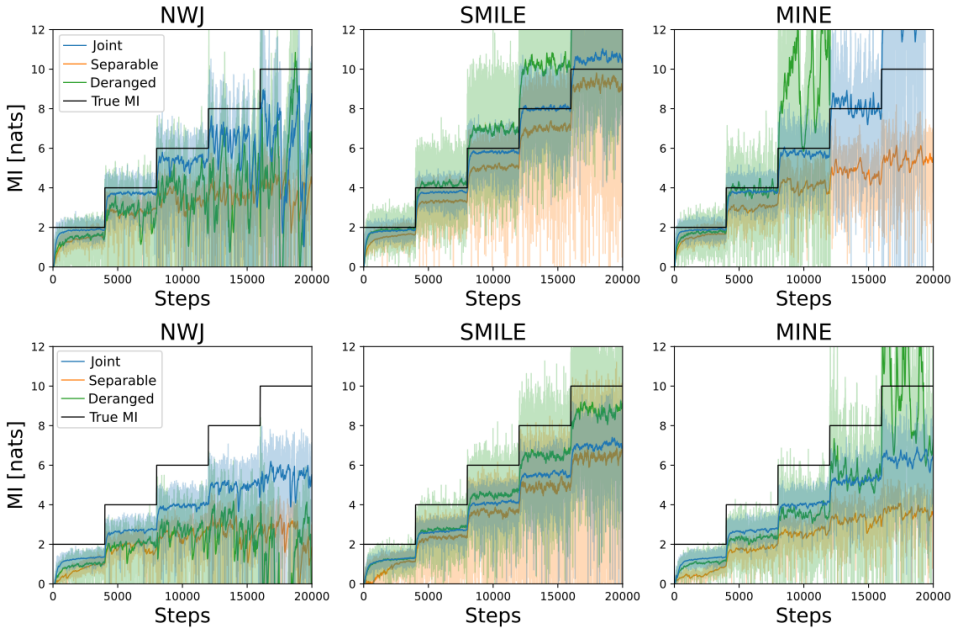


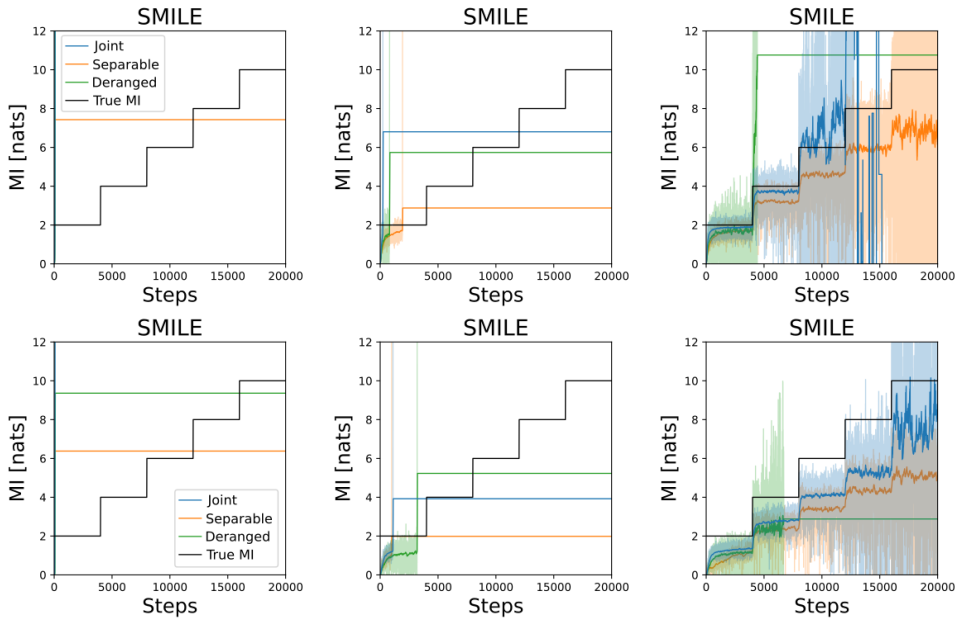
Figure 6: NWJ, SMILE ($\tau = \infty$), and MINE MI estimation comparison with $d = 20$ and $N = 64$. The *Gaussian* setting is represented in the top row, while the *cubic* setting is shown in the bottom row.

Interestingly, by comparing I_{SMILE} ($\tau = \infty$) trained with the JS divergence and with the MINE cost function (in Fig. 6 and Fig. 7, respectively), the variance of the latter case is significantly higher. Hence, the JS maximization trick seems to have an impact in lowering the estimator variance.

D.1.2 Analysis for Different Values of d and N

The class of f-DIME estimators is robust to changes in d and N, as the estimators' variance decreases (see (68) and Fig. 11) when N increases and their achieved bias is not significantly influenced by the choice of d. Differently, I_{NJEE} and I_{CPC} are highly affected by variations of those parameters, as described in Fig. 3 and Fig. 4. More precisely, I_{CPC} is not strongly influenced by a change of d, but the bias significantly increases as the batch size diminishes, since the upper bound lowers. I_{NJEE} achieves a higher bias both when d decreases and when N increases w.r.t. the default values $d = 20, N = 64$. In addition, when d is large, the training of I_{NJEE} is not feasible, as it requires a lot of time (see Fig. 5) and memory (as a consequence of the large number of neural networks utilized) requirements.

We show the achieved bias, variance, and mean squared error (MSE) corresponding to the three settings reported in Fig. 2, 3, and 4 in Fig. 8, 9, and 10, respectively. The achieved variance is bounded when the estimator used is $I_{KL-DIME}$ or I_{CPC}. In particular, Figures 8, 9, 10, and 11 demonstrate that $I_{KL-DIME}$ satisfies Lemma **??**.
Additionally, we report the achieved bias, variance and MSE when $d = 20$ and N varies according to Tab. 1. We use the notation $N = [512, 1024]$ to indicate that each cell of the table reports the values corresponding to $N = 512$ and $N = 1024$, with this specific order, inside the brackets. Similarly, we show the attained bias, variance, and MSE for $d = [5, 10]$ and $N = 64$ in Tab. 3. The achieved bias, variance and MSE shown in Tab. 1 and Tab. 3 motivate that the class of f-DIME estimators attains the best values for bias and MSE. Similarly, $I_{KL-DIME}$ obtains the lowest variance, when excluding I_{CPC} from the estimators comparison (I_{CPC} should not be desirable as it is upper bounded). The illustrated results are obtained with the *joint* architecture (except for NJEE) because, when the batch size is small, such an architecture achieves slightly better results than the *deranged* one, as it approximates the expectation over the product of marginals with more samples.



(a) $\tau = 1$ (b) $\tau = 5$ (c) $\tau = \infty$

Figure 7: I_{SMILE} behavior for different values of τ, when the JS divergence is not used to train the neural network. The *Gaussian* case is reported in the top row, while the *cubic* case is reported in the bottom row.

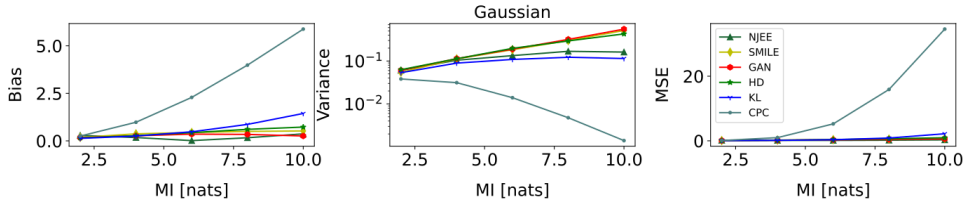


Figure 8: Bias, variance, and MSE comparison between estimators, using the joint architecture for the *Gaussian* case with $d = 20$ and $N = 64$.

The variance of the f-DIME estimators achieved in the Gaussian setting when N ranges from 64 to 1024 is reported in Fig. 11. The behavior shown in such a figure demonstrates what is stated in Lemma **??**, i.e., the variance of the f-DIME estimators varies as $\frac{1}{N}$.

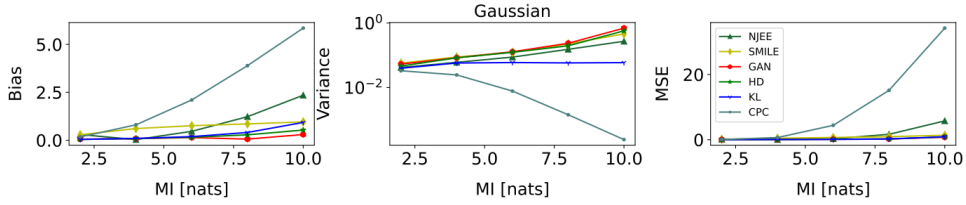


Figure 9: Bias, variance, and MSE comparison between estimators, using the joint architecture for the *Gaussian* case with $d = 5$ and $N = 64$.

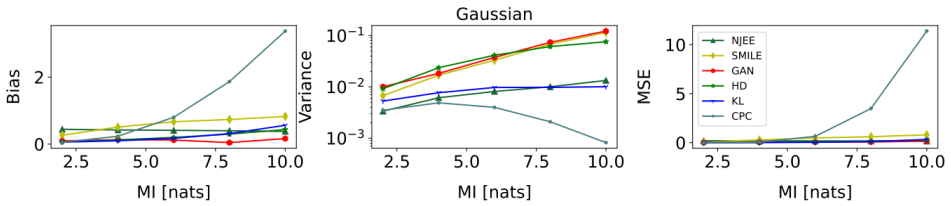


Figure 10: Bias, variance, and MSE comparison between estimators, using the joint architecture for the *Gaussian* case with $d = 20$ and $N = 1024$.

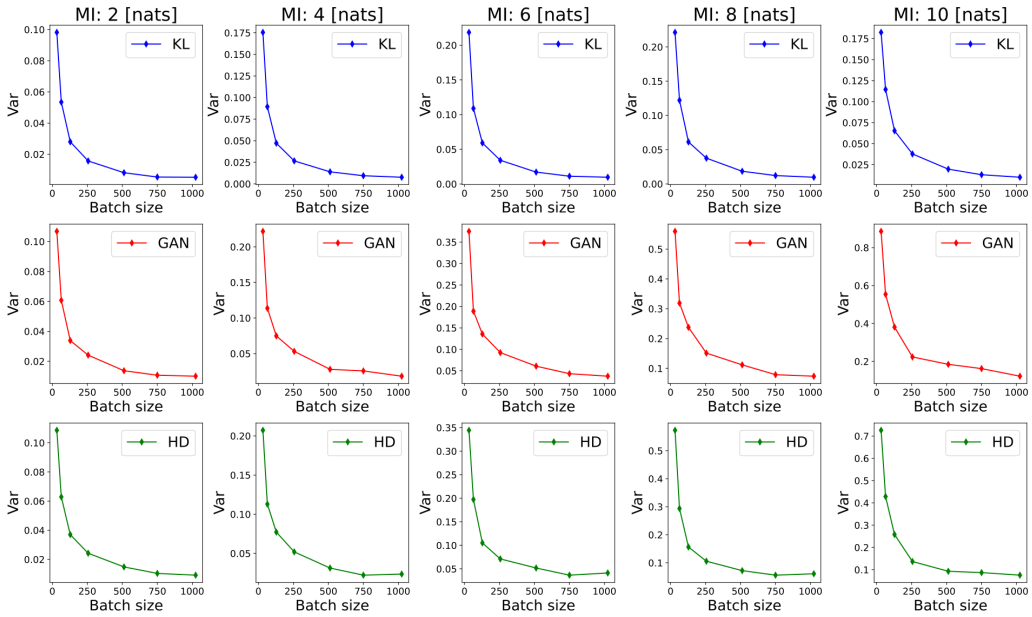


Figure 11: Variance of the f-DIME estimators corresponding to different values of batch size.

Table 1: Bias (B), variance (V), and MSE (M) of the MI estimators using the joint architecture, when $d = 20$ and $N = [512, 1024]$, for the Gaussian setting. Each f-DIME estimator is abbreviated to f-D.

		Gaussian				
	MI	2	4	6	8	10
B	NJEE	[0.42, 0.44]	[0.40, 0.42]	[0.37, 0.41]	[0.34, 0.40]	[0.32, 0.38]
	SMILE	[0.25, 0.27]	[0.48, 0.51]	[0.64, 0.67]	[0.74, 0.73]	[0.86, 0.83]
	GAN-D	[0.11, 0.09]	[0.15, 0.13]	[**0.16**, **0.12**]	[**0.14**, **0.04**]	[**0.01**, **0.16**]
	HD-D	[0.08, 0.07]	[0.15, 0.12]	[0.24, 0.20]	[0.37, 0.30]	[0.47, 0.43]
	KL-D	[**0.07**, 0.06]	[**0.12**, **0.10**]	[0.21, 0.17]	[0.38, 0.31]	[0.69, 0.56]
	CPC	[0.08, **0.05**]	[0.34, 0.23]	[1.07, 0.80]	[2.32, 1.87]	[3.96, 3.37]
V	NJEE	[0.01, 0.00]	[0.01, 0.01]	[0.02, 0.01]	[0.02, 0.01]	[0.02, 0.01]
	SMILE	[0.01, 0.01]	[0.03, 0.02]	[0.06, 0.03]	[0.11, 0.07]	[0.17, 0.11]
	GAN-D	[0.01, 0.01]	[0.03, 0.02]	[0.06, 0.04]	[0.11, 0.07]	[0.17, 0.12]
	HD-D	[0.01, 0.01]	[0.03, 0.02]	[0.05, 0.04]	[0.07, 0.06]	[0.09, 0.08]
	KL-D	[0.01, 0.01]	[0.01, 0.01]	[0.02, 0.01]	[0.02, 0.01]	[0.02, 0.01]
	CPC	[**0.01**, **0.00**]	[**0.01**, **0.00**]	[**0.01**, **0.00**]	[**0.00**, **0.00**]	[**0.00**, **0.00**]
M	NJEE	[0.18, 0.20]	[0.18, 0.18]	[0.16, 0.18]	[0.14, 0.17]	[**0.12**, 0.16]
	SMILE	[0.08, 0.08]	[0.26, 0.28]	[0.47, 0.48]	[0.66, 0.61]	[0.90, 0.80]
	GAN-D	[0.03, 0.02]	[0.05, 0.04]	[0.09, 0.05]	[**0.13**, **0.08**]	[0.18, **0.15**]
	HD-D	[0.02, 0.01]	[0.05, 0.04]	[0.11, 0.08]	[0.21, 0.15]	[0.31, 0.26]
	KL-D	[**0.01**, **0.01**]	[**0.03**, **0.02**]	[**0.06**, **0.04**]	[0.17, 0.11]	[0.49, 0.33]
	CPC	[0.01, 0.01]	[0.13, 0.06]	[1.16, 0.64]	[5.38, 3.48]	[15.67, 11.38]

Table 2: Bias (B), variance (V), and MSE (M) of the MI estimators using the joint architecture, when $d = [5, 10]$ and $N = [64]$, for the Gaussian setting. Each f-DIME estimator is abbreviated to f-D.

		Gaussian				
	MI	2	4	6	8	10
B	NJEE	[0.30, 0.29]	[**0.03**, 0.13]	[0.46, **0.06**]	[1.23, 0.38]	[2.35, 0.80]
	SMILE	[0.29, 0.24]	[0.61, 0.52]	[0.76, 0.68]	[0.85, 0.71]	[0.96, 0.68]
	GAN-D	[0.06, 0.12]	[0.09, 0.17]	[**0.14**, 0.17]	[**0.06**, **0.20**]	[**0.30**, **0.18**]
	HD-D	[0.04, 0.09]	[0.09, 0.14]	[0.15, 0.22]	[0.28, 0.39]	[0.53, 0.40]
	KL-D	[**0.04**, **0.07**]	[0.09, **0.13**]	[0.19, 0.30]	[0.40, 0.58]	[0.93, 1.05]
	CPC	[0.17, 0.20]	[0.80, 0.89]	[2.10, 2.20]	[3.89, 3.93]	[5.85, 5.86]
V	NJEE	[0.04, 0.05]	[0.06, 0.08]	[0.09, 0.10]	[0.15, 0.13]	[0.27, 0.13]
	SMILE	[0.06, 0.06]	[0.09, 0.13]	[0.12, 0.20]	[0.23, 0.32]	[0.46, 0.46]
	GAN-D	[0.05, 0.06]	[0.08, 0.12]	[0.13, 0.19]	[0.24, 0.30]	[0.69, 0.52]
	HD-D	[0.05, 0.06]	[0.08, 0.11]	[0.12, 0.16]	[0.20, 0.24]	[0.57, 0.49]
	KL-D	[0.04, 0.05]	[0.06, 0.08]	[0.06, 0.10]	[0.06, 0.10]	[0.06, 0.10]
	CPC	[**0.03**, **0.04**]	[**0.02**, **0.03**]	[**0.01**, **0.01**]	[**0.00**, **0.00**]	[**0.00**, **0.00**]
M	NJEE	[0.13, 0.13]	[**0.06**, **0.09**]	[0.30, **0.10**]	[1.66, **0.28**]	[5.78, 0.76]
	SMILE	[0.14, 0.11]	[0.46, 0.40]	[0.70, 0.66]	[0.95, 0.83]	[1.37, 0.93]
	GAN-D	[0.06, 0.08]	[0.09, 0.15]	[0.15, 0.22]	[0.24, 0.34]	[**0.78**, **0.55**]
	HD-D	[0.05, 0.07]	[0.09, 0.13]	[0.15, 0.21]	[0.28, 0.40]	[0.86, 0.65]
	KL-D	[**0.04**, **0.06**]	[0.07, 0.10]	[**0.10**, 0.19]	[**0.22**, 0.44]	[0.92, 1.20]
	CPC	[0.06, 0.08]	[0.67, 0.83]	[4.42, 4.84]	[15.14, 15.45]	[34.22, 34.32]

Table 3: Bias (B), variance (V), and MSE (M) of the MI estimators using the joint architecture, when $d = [5, 10]$ and $N = [64]$, for the Gaussian setting. Each f-DIME estimator is abbreviated to f-D.

		Gaussian				
	MI	2	4	6	8	10
B	NJEE	0.29	**0.18**	**0.01**	**0.17**	0.37
	SMILE	0.18	0.37	0.44	0.50	0.52
	GAN-D	0.17	0.27	0.35	0.34	**0.26**
	HD-D	0.16	0.28	0.43	0.61	0.73
	KL-D	**0.13**	0.25	0.48	0.87	1.44
	CPC	0.25	0.98	2.29	3.99	5.88
V	NJEE	0.06	0.10	0.13	0.17	0.16
	SMILE	0.05	0.11	0.18	0.30	0.51
	GAN-D	0.06	0.11	0.19	0.32	0.55
	HD-D	0.06	0.11	0.20	0.29	0.43
	KL-D	0.05	0.09	0.11	0.12	0.11
	CPC	**0.04**	**0.03**	**0.01**	**0.00**	**0.00**
M	NJEE	0.14	**0.14**	**0.13**	**0.20**	**0.30**
	SMILE	0.09	0.25	0.38	0.55	0.79
	GAN-D	0.09	0.19	0.31	0.43	0.62
	HD-D	0.09	0.19	0.39	0.66	0.96
	KL-D	**0.07**	0.15	0.34	0.87	2.19
	CPC	0.10	0.99	5.25	15.89	34.57

The class f-DIME is able to estimate the MI for high-dimensional distributions, as shown in Fig. 12, where $d = 100$. In that figure, the estimates behavior is obtained by using the simple architectures described in Sec. D.1 of the Appendix. Thus, the input size of these neural networks (200) is comparable with the number of neurons in the hidden layers (256). Therefore, the estimates could be improved by increasing the number of hidden layers and neurons per layer. The graphs in Fig. 12 illustrate the advantage of the architecture *deranged* over the *separable* one.

D.1.3 Considerations on Derangements

To facilitate the understanding of the role of derangements during training, we provide a practical example in the following.

Suppose for simplicity that $N = 3$. Then, a random permutation of $\mathbf{y} = [y_1, y_2, y_3]$ can be $[y_2, y_3, y_1]$, where the number of fixed points is $K = 0$ as no elements remain in the same position after the permutation. However, another permutation of $\mathbf{y}$ is $[y_1, y_3, y_2]$. In this case, it is evident that y_1 remains in the same initial position, and the number of fixed points is $K = 1$. A random derangement of $\mathbf{y} = [y_1, y_2, y_3]$, instead, ensures by definition that no element of $\mathbf{y}$ ends up in the same initial position, contrarily from a naive random permutation. This idea is essential to avoid having shuffled marginal samples that actually are realizations of the joint distribution. In fact, we proved that a random permutation strategy would lead to a biased estimator (see the permutation bound in Corollary 2.1).

D.1.4 Time Complexity Analysis

The computational time analysis is developed on a server with CPU "AMD Ryzen Threadripper 3960X 24-Core Processor" and GPU "MSI GeForce RTX 3090 Gaming X Trio 24G, 24GB GDDR6X".
Before analyzing the time requirements to complete the 5-step MI staircases, we specify two different ways to implement the derangement of the $\mathbf{y}$ realizations in each batch:

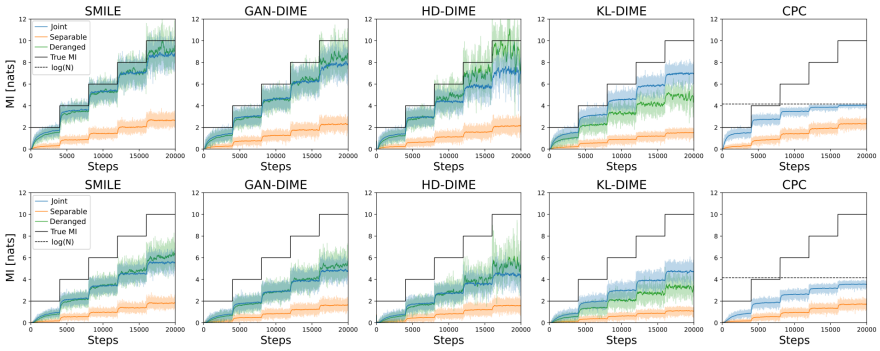


Figure 12: MI estimates when $d = 100$ and $N = 64$. The *Gaussian* setting is represented in the top row, while the *cubic* setting is shown in the bottom row.

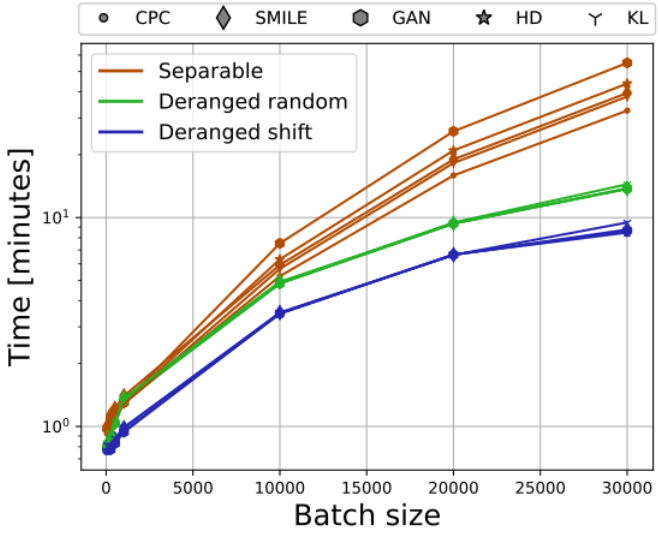


Figure 13: Comparison between the time requirements to complete the 5-step staircases for the architectures *separable*, *deranged* with random-based derangement, and *deranged* with shift-based derangement.

Table 4: Summary of the MI estimators.

Estimator	Low MI		High MI		Scalability
	Bias	Variance	Bias	Variance	
$I_{KL-DIME}$	✓✓	✓✓	$\sim$	✓✓	✓✓
$I_{HD-DIME}$	✓✓	✓✓	✓	✓	✓✓
$I_{GAN-DIME}$	✓✓	✓✓	✓✓	✓	✓✓
$I_{SMILE}(\tau = 1)$	✓	✓✓	✓	✓	✓✓
I_{NJEE}	✓	✓✓	✓	✓✓	✗
I_{CPC}	$\sim$	✓✓	✗	✓✓	✗
$I_{SMILE}(\tau = \infty)$	✓	$\sim$	✓	✗	✓✓
I_{MINE}	✓	✗	✗	✗	✓✓
I_{NWJ}	✓	✗	✗	✗	✓✓

- **Random-based**. The trivial way to achieve the derangement is to randomly shuffle the elements of the batch until there are no fixed points (i.e., all the $\mathbf{y}$ realizations in the batch are assigned to a different position w.r.t. the starting location).

- **Shift-based**. Given N realizations $(\mathbf{x}_i, \mathbf{y}_i)$ drawn from $p_{XY}(\mathbf{x}, \mathbf{y})$, for $i \in \{1, \dots, N\}$, we obtain the deranged samples as $(\mathbf{x}_i, \mathbf{y}_{(i+1)\%N})$, where "%" is the modulo operator.

Although the MI estimates obtained by the two derangement methods are almost indistinguishable, all the results shown in the paper are achieved by using the random-based method. We additionally demonstrate the time efficiency of the shift-based approach.

We show in Fig. 5 that the architectures *deranged* and *separable* are significantly faster w.r.t. *joint* and NJEE ones, for a given batch size N and input distribution size d.
However, Fig. 5 exhibits no difference between the *deranged* and *separable* architectures. Fig. 13 illustrates a detailed representation of the time requirements of these two architectures to complete the 5-step stairs presented in Sec. 6. As N increases, the gap between the time needed by the architectures *deranged* and *separable* grows, demonstrating that the former is the fastest. For example, when $d = 20$ and $N = 30k$, $I_{GAN-DIME}$ needs about 55 minutes when using the architecture *separable*, but only 15 minutes when using the *deranged* one and less than 9 minutes for the shift-based *deranged* architecture.

D.1.5 Summary of the Estimators

We give an insight on how to choose the best estimator in Tab. 4, depending on the desired specifics. We assign qualitative grades to each estimator over different performance indicators. All the indicators names are self-explanatory, except from *scalability*, which describes the capability of the estimator to obtain precise estimates when d and N vary from the default values ($d = 20$ and $N = 64$). The grades ranking is, from highest to lowest: ✓✓, ✓, $\sim$, ✗.
When more than one architecture is available for a specific estimator, the grade is assigned by considering the best architecture within that particular case.
Even though the estimator choice could depend on the specific problem, we consider $I_{GAN-DIME}$ to be the best one. The rationale behind this decision is that $I_{GAN-DIME}$ achieves the best performance for almost all the indicators and lacks weaknesses. Differently, I_{CPC} estimate is upper bounded, I_{SMILE} achieves slightly higher bias, and I_{NJEE} is strongly d and N dependent. However, if the considered problem requires the estimation of a low-valued MI, $I_{KL-DIME}$ is slightly more accurate than $I_{GAN-DIME}$.

One limitation of this paper is that the set of f-divergences analyzed is restricted to three elements. Thus, probably there exists a more effective f-divergence which is not analyzed in this paper.

D.2 Self-Consistency Tests

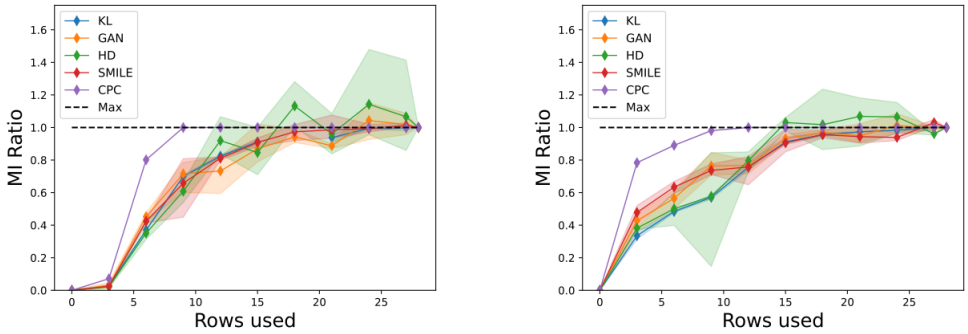
The benchmark considered for the self-consistency tests is similar to the one applied in prior work (Song & Ermon, 2020). We use the images collected in MNIST (LeCun et al., 1998) and FashionMNIST (Xiao et al., 2017) data sets. Here, we test three properties of MI estimators over images distributions, where the MI is not known, but the estimators consistency can be tested:

1. **Baseline**. X is an image, Y is the same image masked in such a way to show only the top t rows. The value of $\hat{I}(X;Y)$ should be non-decreasing in t, and for $t = 0$ the estimate should be equal to 0, since X and Y would be independent. Thus, the ratio $\hat{I}(X;Y)/\hat{I}(X;X)$ should be monotonically increasing, starting from 0 and converging to 1.
2. **Data-processing**. $\bar{X}$ is a pair of identical images, $\bar{Y}$ is a pair containing the same images masked with two different values of t. We set $h(Y)$ to be an additional masking of Y of 3 rows. The estimated MI should satisfy $\hat{I}([X,X];[Y,h(Y)])/\hat{I}(X;Y) \approx 1$, since including further processing should not add information.
3. **Additivity**. $\bar{X}$ is a pair of two independent images, $\bar{Y}$ is a pair containing the masked versions (with equal t values) of those images. The estimated MI should satisfy $\hat{I}([X_1,X_2];[Y_1,Y_2])/\hat{I}(X;Y) \approx 2$, since the realizations of the X and Y random variables are drawn independently.

These tests are developed for I_{fDIME}, I_{CPC}, and I_{SMILE}. Differently, I_{NJEE} training is not feasible, since by construction $2d-1$ models should be created, with $d = 784$ (the gray-scale image shape is 28×28 pixels). The neural network architecture used for these tests is referred to as **conv**.

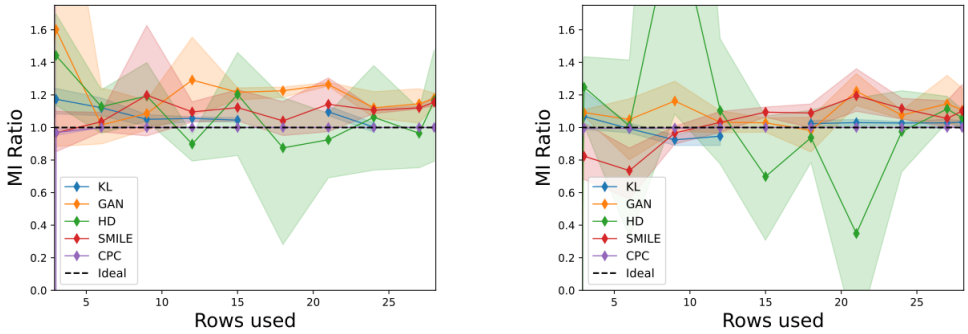
Conv. It is composed by two convolutional layers and one fully connected. The first convolutional layer has 64 output channels and convolves the input images with (5×5) kernels, stride $2\,px$ and padding $2\,px$. The second convolutional layer has 128 output channels, kernels of shape (5×5), stride $2\,px$ and padding $2\,px$. The fully connected layer contains 1024 neurons. ReLU activation functions are used in each layer (except from the last one). The input data are concatenated along the channel dimension. We set the batch size equal to 256.

The comparison between the MI estimators for varying values of t is reported in Fig. 14, 15, and 16. The behavior of all the estimators is evaluated for various random seeds. These results highlight that almost all the analyzed estimators satisfy the first two tests ($I_{HD-DIME}$ is slightly unstable), while none of them is capable of fulfilling the additivity criterion. Nevertheless, this does not exclude the existence of an f-divergence capable to satisfy all the tests.



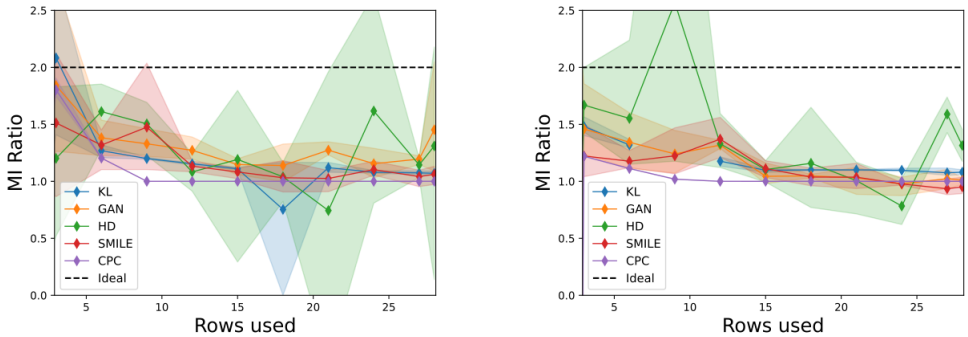
(a) Baseline property, MNIST digits data set. (b) Baseline property, FashionMNIST data set.

Figure 14: Comparison between different estimators for the baseline property, using MNIST data set on the left and FashionMNIST on the right.



(a) Data processing property, MNIST digits data set. (b) Data processing property, FashionMNIST data set.

Figure 15: Comparison between different estimators for the data processing property, using MNIST data set on the left and FashionMNIST on the right.



(a) Additivity property, MNIST digits data set. (b) Additivity property, FashionMNIST data set.

Figure 16: Comparison between different estimators for the additivity property, using MNIST data set on the left and FashionMNIST on the right.