Neural Mutual Information Estimation with Vector Copulas

Yanzhi Chen[1], **Zijing Ou**[2], **Adrian Weller**[1,3], **Michael U. Gutmann**[4]
[1]University of Cambridge, [2]Imperial College London, [3]Alan Turing Institute, [4]University of Edinburgh

Abstract

Estimating mutual information (MI) is a fundamental task in data science and machine learning. Existing estimators mainly rely on either highly flexible models (e.g., neural networks), which require large amounts of data, or overly simplified models (e.g., Gaussian copula), which fail to capture complex distributions. Drawing upon recent vector copula theory, we propose a principled interpolation between these two extremes to achieve a better trade-off between complexity and capacity. Experiments on state-of-the-art synthetic benchmarks and real-world data with diverse modalities demonstrate the advantages of the proposed estimator.

1 Introduction

Mutual information (MI) is a fundamental measure of the statistical dependence between random variables (RVs). Compared to other dependence measures, MI stands out due to its equitability and generality [1, 2]: it can capture non-linear dependence of any form and can handle RVs with any dimensionalities, rendering it a powerful measure for quantifying statistical dependence. In data science, MI is widely used to analyze the relationships between protein sequences [3] and gene profiles [4, 5], as well as to assess feature importance and redundancy [6]. In machine learning, MI broadly serves as a learning objective and regularizer [7, 8, 9, 10, 11, 12], with diverse applications to representation learning [7, 8, 9, 13, 14], generative modeling [10], fairness and privacy [15, 16], etc.

A wide range of powerful, neural MI estimators have been developed [17, 18, 19, 20, 21, 22, 23]. Most of these estimators rely on a *single, unconstrained* network to approximate certain quantities—such as the joint density $p(\mathbf{x}, \mathbf{y})$ or the density ratios $p(\mathbf{x}, \mathbf{y})/p(\mathbf{x})p(\mathbf{y})$—during MI estimation. While neural networks as universal functional approximators can, in theory, approximate arbitrary functions given sufficient data [24, 25], in practice we often only have a small set of data. Indeed, theoretical studies have shown that such *distribution-free* treatment of MI estimation will inevitably suffer from requiring an exponential sample size [26, 27, 28, 29]. A straightforward remedy is to restrict the model to simpler classes—for instance, assuming that the data is approximately Gaussian. However, these assumptions are often overly simplistic to capture complex distributions in reality.

Recent advances in vector copula theory [30] offer a promising avenue for addressing this dilemma. Vector copula theory extends classical copula theory [31] by generalizing it from *univariate* to *vector* marginals. It reveals that the multivariate marginals and the dependence structure (i.e., the vector copula) of a joint distribution are fully disentangled. This disentanglement motivates a more fine-grained way for making assumption in MI estimation, where we impose lightweight yet reasonable assumptions solely on the *vector copula* rather than on the *entire distribution*. Crucially, the complexity of the vector copula can be adaptively adjusted through efficient vector copula selection, allowing for an optimal trade-off between capacity and complexity. Experiments on state-of-the-art synthetic benchmarks and real-world data demonstrate the competitiveness of our estimator against state-of-the-art estimators. In summary, the main contributions of this work are three-fold:

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

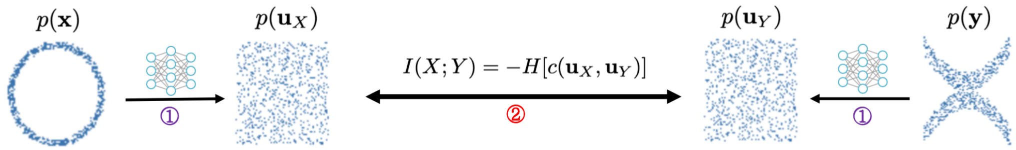


Figure 1: **Overview of the proposed vector copula-based MI estimator** (VCE), which explicitly disentangles the modeling of marginal distribution and dependence structure (i.e., the vector copula). VCE first respectively computes the vector ranks $\mathbf{u}_X$ and $\mathbf{u}_Y$ corresponding to the two marginal variables X and Y with flow models (①). It then finds the vector copula $c \in \mathcal{C}$ from the vector copula pool $\mathcal{C}$ that best matches with the joint distribution $p(\mathbf{u}_X, \mathbf{u}_Y)$ of the estimated vector ranks (②). Mutual information $I(X;Y)$ is computed as the negative differential entropy of the vector copula c, which itself is irrelevant to the two marginal distributions $p(\mathbf{x})$ and $p(\mathbf{y})$.

- We develop a divide-and-conquer MI estimator based on recent vector copula theory, which explicitly disentangles marginal distributions and dependence structure in MI estimation;
- We reinterpret existing estimators through the lens of vector copula, revealing that they correspond to varying parameterization and learning strategies of vector copula with various trade-offs;
- We provide consistency and error analysis of our estimator, along with extensive empirical evaluation on diverse test cases covering multiple modalities, marginal patterns and dependence structures.

Code containing both our method and state-of-the-art neural estimators is available in [github repo].

2 Preliminaries

Throughout this work, we use upper case letters (e.g. X) to denote random variables and lower case letters (e.g. $\mathbf{x}$) to denote their instances. We use $\mathcal{U}[0,1]^d$ or μ to denote the uniform distribution on $[0,1]^d$ and use $\mathcal{N}$ to denote Gaussian distribution on $\mathbb{R}^d$. ∇ denotes the gradient and $J_\mathbf{x}\mathbf{y}$ denotes the Jacobian of $\mathbf{y}$ w.r.t $\mathbf{x}$. The symbol $\#$ denotes the push-forward operation.

2.1 Mutual information and its estimation

The mutual information (MI) between variables X and Y is defined as the Kullback-Leibler (KL) divergence between the joint distribution $p(\mathbf{x}, \mathbf{y})$ and the product of marginal distributions $p(\mathbf{x})p(\mathbf{y})$:

$$I(X;Y) = KL[p(\mathbf{x}, \mathbf{y}) \| p(\mathbf{x})p(\mathbf{y})] = \mathbb{E}\left[\log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}\right] \quad (1)$$

In this work, we consider estimating $I(X;Y)$ from an empirical dataset $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^n$. Several neural network-based methods have been developed for MI estimation:

Generative estimators. These methods leverage generative models to approximate the various distributions in (1) or their equivalents, and use the learned generative models to construct an MI estimate [17, 29, 18, 32, 19]. The accuracy of generative estimators crucially depends on the quality of the learned generative models. Simpler models (e.g. Gaussian copula) are easy to learn but may fail to adequately capture the true data distribution [33, 34]. In contrast, complex models (e.g. flow-based models [35, 36, 37, 38] and diffusion models [19]) offer greater expressiveness but can be challenging to optimize, in particular if the amount of data is insufficient or the data dimensionality is high.

Discriminative estimators. These methods train a neural network f with samples $\mathbf{x}, \mathbf{y} \sim p(\mathbf{x}, \mathbf{y})$ and samples $\mathbf{x}, \mathbf{y} \sim p(\mathbf{x})p(\mathbf{y})$ to estimate the density ratio $p(\mathbf{x}, \mathbf{y})/p(\mathbf{x})p(\mathbf{y})$ [20, 21, 39, 40, 41, 32, 23]. Once trained, the learned density ratio can either be used in (1) or in the Donsker-Varadhan (DV) representation [42] to obtain an MI estimate. Discriminative methods avoid directly modeling densities, however they are prone to the curse of *high-discrepancy* [40, 41, 29, 32], which occurs if $p(\mathbf{x}, \mathbf{y})$ and $p(\mathbf{x})p(\mathbf{y})$ differ significantly — for instances, cases with high MI or high-dimensional data. Several advanced methods were proposed to alleviate this issue, including clipping the network outputs [32], introducing reference distributions [41], avoiding computing the partition function [23].

2.2 Vector copula

Vector copula theory. The recent vector copula theory [30] provides a principled framework for modeling and analyzing the dependence between *multivariate* random variables. It extends classical copula theory [31] by considering 'vector' marginals. We begin by the concept of vector ranks:

Definition 1 (Vector rank)**.** *Let p be an absolutely continuous distribution on $\mathbb{R}^d$ with support in a convex set. Let μ be the uniform distribution on $[0,1]^d$. There exists a convex function ψ such that $\nabla\psi\#\mu = p$ and $\nabla\psi^{-1}\#p = \mu$. $\mathbf{u} := \nabla\psi^{-1}$ is called the vector rank associated with p [43].*

When $d = 1$, vector rank reduces to standard scalar rank. Intuitively, vector rank transforms a multivariate distribution p to a (multivariate) uniform distribution μ, entirely removing its characteristics.

In the text below, we slightly overload this definition and use the term 'vector rank' to refer to both the vector rank function $\mathbf{u}(\cdot)$ and also the corresponding random variable $\mathbf{u}$ induced by this function.

Definition 2 (Vector copula)**.** *Let $\mathbf{u}_X$ and $\mathbf{u}_Y$ be the vector ranks corresponding to $p(\mathbf{x})$ and $p(\mathbf{y})$ respectively. A vector copula $C(\mathbf{u}_X, \mathbf{u}_Y)$ is a cumulative distribution function on $[0,1]^{d_X+d_Y}$ with uniform marginals on $C(\mathbf{u}_X) = \mathcal{U}[0,1]^{d_X}$ and $C(\mathbf{u}_Y) = \mathcal{U}[0,1]^{d_Y}$. The probabilistic density function corresponding to C is called* vector copula density *and is denoted as $c(\mathbf{u}_X, \mathbf{u}_Y)$ [30].*

Given the above definition, we have the following result [30] generalizing the Sklar theorem [31].

Theorem 1 (Vector Sklar Theorem)**.** *Let $X \in \mathbb{R}^{d_X}$ and $Y \in \mathbb{R}^{d_Y}$ be two random variables with joint distribution $p(\mathbf{x}, \mathbf{y})$ on $\mathbb{R}^{d_X+d_Y}$. For any absolutely continuous distributions $p(\mathbf{x}, \mathbf{y})$ with support in a convex set, there exist an unique function $c(\cdot, \cdot)$, such that*

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})c(\mathbf{u}_X, \mathbf{u}_Y) \tag{2}$$

where $\mathbf{u}_X$ and $\mathbf{u}_Y$ are the vector ranks computed for $\mathbf{x}$ and $\mathbf{y}$ respectively. The function c equals to the vector copula density associated with $\mathbf{u}_X$ and $\mathbf{u}_Y$ [30].

The vector Sklar theorem suggests that for a distribution $p(\mathbf{x}, \mathbf{y})$, its marginal distributions and the joint dependence structure are entirely disentangled, with the latter fully characterized by the vector copula density c. Note that here we focus on the case of two RVs; we refer to [30] for general cases.

Instances of vector copula. We discuss several instances of vector copula related to our work. One important instance is the *vector Gaussian copula* [30]. This model assumes that the joint dependence structure admits a Gaussian structure, with its vector copula $C^{\mathcal{N}}$ being

$$C^{\mathcal{N}}(\mathbf{u}_X, \mathbf{u}_Y) = \Phi(\phi^{-1}(\mathbf{u}_X), \phi^{-1}(\mathbf{u}_Y); \mathbf{0}, \Sigma) \tag{3}$$

where $\Sigma = [[\mathbf{I}_X, \Sigma_{XY}], [\Sigma_{XY}^\top, \mathbf{I}_Y]]$ is a p.s.d matrix whose blocks $\mathbf{I}_X \in \mathbb{R}^{d_X \times d_X}$ and $\mathbf{I}_Y \in \mathbb{R}^{d_Y \times d_Y}$ are identity matrices. $\Phi(\cdot)$ is the cumulative distribution function of multivariate normal distribution and $\phi(\cdot)$ is the (element-wise) cumulative distribution function of univariate normal distribution. Equivalently, a vector Gaussian copula can be defined by its data generation process: $\epsilon \sim \mathcal{N}(\epsilon; \mathbf{0}, \Sigma)$, $\mathbf{u}_X = \phi(\epsilon_{\leq d_X})$, $\mathbf{u}_Y = \phi(\epsilon_{> d_X})$, with $\epsilon_{\leq d_X}$ and $\epsilon_{> d_X}$ being the first d_X and the remaining dimensions of ϵ respectively. An analytic expression for $c^{\mathcal{N}}$ can be derived accordingly.

Other useful instances of vector copula include t-vector copula, Archimedean vector copula and Kendall vector copula, which correspond to different inductive biases about the dependence structure.

3 Methodology

In this section, we propose a new mutual information (MI) estimator based on vector copula theory. The core of our method is Theorem 2, which establishes a connection between MI and vector copula:

Theorem 2 (MI is vector copula entropy)**.** *The mutual information $I(X;Y)$ is the negative differential entropy of the vector copula density:*

$$I(X;Y) = -H[c(\mathbf{u}_X, \mathbf{u}_Y)] \tag{4}$$

where $\mathbf{u}_X$ and $\mathbf{u}_Y$ are the vector ranks corresponding to $p(\mathbf{x})$ and $p(\mathbf{y})$ respectively.

Proof: Please refer to Appendix A. □

Algorithm 1 Vector copula MI estimate (VCE)

```
Input: data $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{n}$
Output: estimated $\hat{I}(X;Y)$
Parameters: flows $f_X, f_Y$, copulas $\{c_1, ..c_M\}$
Initialization: $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{val}$, $K = 1$,

▷ Marginal distributions learning
learn $f_X$ with $\mathcal{D}_X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$ by FM;
learn $f_Y$ with $\mathcal{D}_Y = \{\mathbf{y}^{(i)}\}_{i=1}^{n}$ by FM;
for $i$ in 1 to $n$ do
    compute $\hat{\mathbf{u}}_X^{(i)} = \texttt{rank}(f_X(\mathbf{x}^{(i)}))$;
    compute $\hat{\mathbf{u}}_Y^{(i)} = \texttt{rank}(f_Y(\mathbf{y}^{(i)}))$;
end for

▷ Vector copula density estimation
repeat
    set $c(\mathbf{u}_X, \mathbf{u}_Y) = \frac{1}{K}\sum_{k=1}^{K} p_k c_k(\mathbf{u}_X, \mathbf{u}_Y)$;
    $\hat{c} = \arg\max_c \mathbb{E}_{\hat{\mathbf{u}}_X, \hat{\mathbf{u}}_Y \sim \mathcal{D}_{train}}[\log c(\hat{\mathbf{u}}_X, \hat{\mathbf{u}}_Y)]$;
    $\mathcal{L}_{\text{val}} \leftarrow \mathbb{E}_{\hat{\mathbf{u}}_X, \hat{\mathbf{u}}_Y \sim \mathcal{D}_{val}}[\log c(\hat{\mathbf{u}}_X, \hat{\mathbf{u}}_Y)]$;
    $K \leftarrow 2K$;
until no improvement on $\mathcal{L}_{\text{val}}$
return $\hat{I}(X;Y) = \frac{1}{n}\sum_{i=1}^{n} \log \hat{c}(\hat{\mathbf{u}}_X^{(i)}, \hat{\mathbf{u}}_Y^{(i)})$
```

Algorithm 2 Vector copula MI estimate' (VCE')

```
Input: data $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^{n}$
Output: estimated $\hat{I}(X;Y)$
Parameters: flows $f_X$, $f_Y$, ratio estimator $r$
Initialization: reference copula $c'$, $\mathcal{D}' = \emptyset$

▷ Marginal distributions learning
learn $f_X$ with $\mathcal{D}_X = \{\mathbf{x}^{(i)}\}_{i=1}^{n}$ by FM;
learn $f_Y$ with $\mathcal{D}_Y = \{\mathbf{y}^{(i)}\}_{i=1}^{n}$ by FM;
for $i$ in 1 to $n$ do
    compute $\hat{\mathbf{u}}_X^{(i)} = \texttt{rank}(f_X(\mathbf{x}^{(i)}))$;
    compute $\hat{\mathbf{u}}_Y^{(i)} = \texttt{rank}(f_Y(\mathbf{y}^{(i)}))$;
end for

▷ Vector copula density estimation
repeat
    sample $\mathbf{u}_X^{(j)}, \mathbf{u}_Y^{(j)} \sim c'(\mathbf{u}_X, \mathbf{u}_Y)$;
    $\mathcal{D}' \leftarrow \mathcal{D}' \cup \{\mathbf{u}_X^{(j)}, \mathbf{u}_Y^{(j)}\}$;
until $|\mathcal{D}'| = n$
train $r$ to classify samples from $\mathcal{D}$ and $\mathcal{D}'$;
set $\hat{c}(\mathbf{u}_X, \mathbf{u}_Y) = r(\mathbf{u}_X, \mathbf{u}_Y) \cdot c'(\mathbf{u}_X, \mathbf{u}_Y)$;
return $\hat{I}(X;Y) = \frac{1}{n}\sum_{i=1}^{n} \log \hat{c}(\hat{\mathbf{u}}_X^{(i)}, \hat{\mathbf{u}}_Y^{(i)})$
```

This theorem generalizes the results of [44, 45] from univariate to vector marginals[1]. It establishes that MI depends solely on the vector copula, which itself is invariant to marginal distributions. Notably, the theorem also reveals that the pointwise mutual information (PMI) i.e. $p(\mathbf{x}, \mathbf{y})/p(\mathbf{x})p(\mathbf{y})$ can equivalently be viewed as a *density* $c(\mathbf{u}_X, \mathbf{u}_Y)$ in its own right, in contrast to the vast majority of existing works [46, 47, 48, 49] which continue to treat PMI as a *density ratio*. This shift in perspectives opens us new possibility in the parameterization and learning of the PMI, including directly modeling it as a normalized density learned via MLE, as will be discussed later.

Theorem 2 immediately suggests a new *divide-and-conquer* approach for MI estimation: we can first estimate the vector ranks $\mathbf{u}_X$ and $\mathbf{u}_Y$, followed by subsequent learning of the vector copula c[2]:

$$I(X;Y) \approx \hat{I}(X;Y) := \frac{1}{n}\sum_{i=1}^{n} \log \hat{c}(\hat{\mathbf{u}}_X^{(i)}, \hat{\mathbf{u}}_Y^{(i)}) \tag{5}$$

where $\hat{\mathbf{u}}_X$, $\hat{\mathbf{u}}_Y$ and $\hat{c}$ are the empirical estimates to $\mathbf{u}_X$, $\mathbf{u}_Y$ and c respectively.

We discuss below several potential advantages of the above divide-and-conquer estimation strategy:

- By disentangling the *modeling* of marginals distribution and copula, we can use differently-sized models in their parameterization, avoiding using a single overly flexible or overly simplified model for the entire distribution. This leads to a better trade-off between model complexity and capacity;
- By disentangling the *learning* of marginals and copula, we can reuse the pre-trained marginals across multiple copula choices with varying complexities, allowing model selection to be performed solely in the copula space in a computational efficient way. It also reduces overall learning difficulty.

In the following, we elaborate methods to estimate the vector ranks and the vector copula respectively.

[1] Building upon classic copula, the theory in [44, 45] only holds for bivariate cases, and generalizing their results to high-dimensional cases require non-trivial formulation and derivation—precisely our key contribution.

[2] Alternatively, one may also learn the marginals $\hat{p}(\mathbf{x}), \hat{p}(\mathbf{y})$ and the vector copula $\hat{c}$ jointly. However, joint learning can be ill-posed [50]. Our ablation study in Appendix B2 suggests that separate learning is more robust.

3.1 Marginal distribution learning

In this step, we learn the two marginal distributions $p(\mathbf{x})$ and $p(\mathbf{y})$ with flexible flow-based models [35, 36, 37, 38] and use them to compute the vector ranks $\mathbf{u}_X$ and $\mathbf{u}_Y$.

Flow-based modeling of marginals. Let $f_X : \mathbb{R}^{d_X} \to \mathbb{R}^{d_X}$ and $f_Y : \mathbb{R}^{d_Y} \to \mathbb{R}^{d_Y}$ be two flow-based models and let $p_{f_X}(\mathbf{x})$ and $p_{f_Y}(\mathbf{y})$ be the densities induced by f_X and f_Y respectively. We respectively learn f_X and f_Y with data $\mathbf{x} \sim p(\mathbf{x})$ and data $\mathbf{y} \sim p(\mathbf{y})$ by flow matching [38]:

$$\min_{f_X} \mathbb{E}[\mathcal{L}_{\text{FM}}(\mathbf{x}; f_X)], \qquad \min_{f_Y} \mathbb{E}[\mathcal{L}_{\text{FM}}(\mathbf{y}; f_Y)] \tag{6}$$

where $\mathcal{L}_{\text{FM}}$ denotes the flow-matching loss [38]. Upon convergence, f_X and f_Y respectively transform the two marginals to a standard normal distribution: $\mathcal{N}(\mathbf{0}, \mathbf{I}) \approx f_X \# p(\mathbf{x})$ and $\mathcal{N}(\mathbf{0}, \mathbf{I}) \approx f_Y \# p(\mathbf{y})$.

Vector ranks computation. With the learned flows f_X and f_Y, we compute the vector ranks as:

$$\hat{\mathbf{u}}_X^{(i)} = \texttt{rank}(f_X(\mathbf{x}^{(i)})), \qquad \hat{\mathbf{u}}_Y^{(i)} = \texttt{rank}(f_Y(\mathbf{y}^{(i)})) \tag{7}$$

where $\texttt{rank}_d(\boldsymbol{\epsilon}) = \frac{1}{n+1} \sum_{j=1}^{n} \mathbf{1}[\epsilon_d \geq \epsilon_d^{(j)}]$ is the element-wise ranking function that computes the scalar ranks for each of the dimension in ϵ. Given universal density approximators f_X, f_Y, $\hat{\mathbf{u}}_X$ and $\hat{\mathbf{u}}_Y$ serve as consistent estimates of the true vector ranks $\mathbf{u}_X$ and $\mathbf{u}_Y$.

While the joint density $p(\mathbf{x}, \mathbf{y})$ is often challenging to estimate, the marginal distributions $p(\mathbf{x})$ and $p(\mathbf{y})$ are typically far easier to learn due to their lower dimensionality. It is thus reasonable to expect that $\hat{\mathbf{u}}_X$ and $\hat{\mathbf{u}}_Y$ are close approximations to $\mathbf{u}_X$ and $\mathbf{u}_Y$ in moderate dimensionality settings.

Remark. The above process of estimating $\mathbf{u}_X$ and $\mathbf{u}_Y$ can be viewed as a generalization of classic copula transformation in MI estimation, where we compute vector ranks rather than scalar ranks.

3.2 Vector copula estimation

In this step, we learn the vector copula c with the previously estimated vector ranks $\hat{\mathbf{u}}_X$ and $\hat{\mathbf{u}}_Y$, leveraging a model-based parameterization and a careful model selection strategy.

Model-based parameterization of copula. As noted earlier, any parametric model can be used to represent the vector copula c, regardless of whether an analytical PMI is available. In this work, we parameterize c as a mixture of existing parametric vector copulas [30] from the copula pool, whose model complexity can be well controlled by tuning the number of mixture components:

$$c(\mathbf{u}_X, \mathbf{u}_Y) = \sum_{k=1}^{K} p_k c_k(\mathbf{u}_X, \mathbf{u}_Y), \tag{8}$$

where $\sum_{k=1}^{K} p_k = 1$ and each $c_k \in \mathcal{C}$ is selected from the predefined pool $\mathcal{C}$ of vector copulas. Any inductive bias about the dependence structure can be used to guide copula selection. Here, we simply implement each c_k as a vector Gaussian copula and learn c by maximum likelihood estimate (MLE):

$$\max_{c} \mathbb{E}[\log c(\mathbf{u}_X, \mathbf{u}_Y)] \tag{9}$$

In theoretical analysis, we analyze why this copula design is a cheap yet reasonable modeling of c.

Efficient model selection. A key design in our method is the explicit exploration of the capacity–complexity trade-off in copula modeling, which is governed by the number of mixture components K. Here, we determine K by cross validation, using negative log-likelihood (NLL) as the criterion. This process is computationally cheap: each copula is already lightweight, involving no neural networks; furthermore, different copulas can be trained in parallel using one single loss.

Algorithm 1 summarizes the main pipeline of the proposed vector copula-based estimator (VCE).

Remark. As an alternative to the above model-based parameterization, one may also adopt a reference-based parameterization for the vector copula, inspired by the design in [51]. Specifically, let c' be a reference vector copula that is easy to sample (e.g. a vector Gaussian copula). We can learn c by first estimating the density ratio $r = c/c'$ using samples from c and c' [52, 41, 39], then recover the vector copula c as $c = r \cdot c'$; see Algorithm 2. By parameterizing r as a deep neural network, this method allows for a more flexible modeling of c, at the cost of a less fine-grained control over its complexity.

4 Theoretical analysis

In this section, we analyze several important theoretical properties of the proposed VCE estimator.

Proposition 1 (Consistency of VCE)**.** *Assuming that (a) f_X and f_Y are universal PDF approximators with continuous support and (b) the number of mixture components K in* (8) *is sufficiently large. Define $\hat{I}_n(X;Y) := \frac{1}{n}\sum_{i=1}^{n} \log \hat{c}(\hat{\mathbf{u}}_X^{(i)}, \hat{\mathbf{u}}_Y^{(i)})$. For every $\epsilon > 0$, there exists $n(\varepsilon) \in \mathbb{N}$, such that*

$$\left|\hat{I}_n(X;Y) - I(X;Y)\right| < \varepsilon, \quad \forall n \geq n(\varepsilon), a.s.$$

Proof. Please refer to Appendix A □

Additionally, we have the following result analyzing the estimation error w.r.t the quality of the learned marginals $p_{f_X}(\mathbf{x}), p_{f_Y}(\mathbf{y})$ and the estimated vector copula density $\hat{c}$.

Proposition 2 (Error of vector copula-based MI estimate)**.** *Let $\hat{\mathbf{u}}_X$ and $\hat{\mathbf{u}}_Y$ be the estimated vector ranks. Let $c(\hat{\mathbf{u}}_X, \hat{\mathbf{u}}_Y)$ be the true joint distribution of $\hat{\mathbf{u}}_X$ and $\hat{\mathbf{u}}_Y$, and $\hat{c}(\hat{\mathbf{u}}_X, \hat{\mathbf{u}}_Y)$ its estimate. Assuming that sufficient Monte Carlo samples are used to compute $\hat{I}(X;Y)$ in* (5)*, we have*

$$\left|I(X;Y) - \hat{I}(X;Y)\right| \leq \left|H(\hat{\mathbf{u}}_X) + H(\hat{\mathbf{u}}_Y)\right| + KL[c(\hat{\mathbf{u}}_X, \hat{\mathbf{u}}_Y))\|\hat{c}(\hat{\mathbf{u}}_X, \hat{\mathbf{u}}_Y)]$$

where the first term on the RHS vanishes as $p_{f_X}(\mathbf{x}) \to p(\mathbf{x})$ and $p_{f_Y}(\mathbf{y}) \to p(\mathbf{y})$. In the limit of perfectly learned marginals, the error simplifies to

$$|I(X;Y) - \hat{I}(X;Y)| = KL[c\|\hat{c}],$$

with c and $\hat{c}$ being the true vector copula density and estimated vector copula density, respectively.

Proof. Please refer to Appendix A. □

Proposition 2 decomposes the estimation error of the proposed VCE estimator into two components:

- *Marginal estimation error*. Imperfect marginal estimations introduce a bias given by $|H(\hat{\mathbf{u}}_X) + H(\hat{\mathbf{u}}_Y)| > 0$, which diminishes as both marginals are learned more accurately (recall that ideally, we have $\hat{\mathbf{u}}_X \sim \mathcal{U}[0,1]^{d_X}$ and $\hat{\mathbf{u}}_Y \sim \mathcal{U}[0,1]^{d_Y}$). For data with moderate dimensionality, we expect this bias to be small, as the two marginals are with low-dimensionality, being easy to estimate.
- *Dependence structure modeling error*. This error arises from the discrepancy between the estimated copula $\hat{c}$ and the true copula c. It depends on two factors: (a) *capacity* - whether the parameterization of $\hat{c}$ is sufficiently expressive to approximate c; and (b) *complexity* - how easy $\hat{c}$ can be learned from the limited data. These factors highlight the importance of model selection for the copula c.

Proposition 3 (Vector Gaussian copula as second-order approximation)**.** *A vector Gaussian copula $c^{\mathcal{N}}$ corresponds to the second-order Taylor expansion of the true vector copula c^* up to variable transformation.*

Proof. Please refer to Appendix A. □

This result explains our choice of using a mixture of Gaussian copulas as a cheap yet principled approximation to the true vector copula. A single vector Gaussian copula already offers a reasonable approximation of the true copula by capturing dependencies up to second order; higher-order interactions, if necessary, can be modeled by adding mixture components in a fully controllable way.

Finally, we have the following result regarding cases with weakly dependent random variables (RVs).

Proposition 4 (Vector copula of independent RVs)**.** *The vector copula corresponding to the product of marginals $p'(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})$ is a vector Gaussian copula if $p'(\mathbf{x}, \mathbf{y})$ is absolutely continuous.*

Proof. Please refer to Appendix A. □

Proposition 4 suggests that if the two RVs X and Y are nearly independent, our estimator is *likely* to provide an accurate estimation of $I(X;Y)$ as the true vector copula is Gaussian-like, being close to the family of our copula design (8). For weakly dependent RVs, it is reasonable to expect that $p(\mathbf{x}, \mathbf{y})$ resembles a vector Gaussian copula, with the difference captured by the additional components in (8).

5 Reinterpreting existing MI estimators

In this section, we reinterpret existing MI estimators through the lens of vector copula theory, showing that they correspond to different parameterizations and learning strategies of the vector copula.

Reinterpreting discriminative estimators. Existing critic-based approach to MI estimation [20, 53, 39, 41, 32, 23] can be interpreted as parameterizing the vector copula $c(\mathbf{u}_X, \mathbf{u}_Y)$ using a feedforwarding neural network f:

$$c(\mathbf{u}_X, \mathbf{u}_Y) \propto e^{f(\mathbf{x},\mathbf{y})} \tag{10}$$

which is learned by discerning samples from the joint $p(\mathbf{x}, \mathbf{y})$ and the product of marginals $p(\mathbf{x})p(\mathbf{y})$ (via e.g contrastive learning). Specifically, recall that the optimal critic f in these methods corresponds to the log density ratio up to an additive constant C [54]: $f(\mathbf{x}, \mathbf{y}) = \log p(\mathbf{x}, \mathbf{y})/p(\mathbf{x})p(\mathbf{y}) + C$, with the PMI itself equal to the vector copula density, as established by the vector Skalar theorem.

Compared to our model-based parameterization of the vector copula density in (8), this neural network parameterization is more flexible and can potentially capture more complex dependence structures. However, as discussed earlier, such distribution-free parameterizations lack complexity control, which may lead to a poor bias–variance trade-off. Furthermore, discriminative methods learns the vector copula by comparing distributions, which can be challenging if they differ significantly (e.g., in high-MI cases, see [40, 41, 29, 32]. In contrast, our main method learns the vector copula by maximum likelihood estimate (MLE), which is the most efficient consistent estimator for the copula.

Reinterpreting generative estimators. Many generative estimators for MI [17, 29, 18, 48, 32] require either learning the joint distribution $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y})c(\mathbf{u}_X, \mathbf{u}_Y)$ or the conditional distribution $p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y})c(\mathbf{u}_X, \mathbf{u}_Y)$ using a single model. This process can be interpreted as learning the marginal distribution(s) and the vector copula simultaneously, with the two components parameterized *jointly* via a *single* generative model. Our method, on the contrary, explicitly separates the modeling and the learning of the marginal distributions $p(\mathbf{x}), p(\mathbf{y})$ from that of the vector copula $c(\mathbf{u}_X, \mathbf{u}_Y)$. This strategy not only enables a more fine-grained control over model complexity, but also mitigates the challenge of jointly learning the marginal distribution and the dependence structure—a strategy aligned with the spirit of classical copula transformations [47, 55, 56, 57] to simplify MI estimation.

We further discuss two recent works [48, 17] closely related to our work. These methods operate by respectively transforming the two RVs X and Y by two flow-based models, such that the joint distribution of the transformed data can be approximated by a distribution with an easy-to-compute MI (for instance, a Gaussian distribution). Their practical methods, $\mathcal{N}$-MIENF and DINE-Gaussian, can be reinterpreted as assuming the dependence structure as a vector Gaussian copula (see Lemma 3 in Appendix A4 for a detailed derivation):

$$c(\mathbf{u}_X, \mathbf{u}_Y) \approx c^{\mathcal{N}}(\mathbf{u}_X, \mathbf{u}_Y; \Sigma) \tag{11}$$

which corresponds to the case $K = 1$ in the VCE estimator and is accurate (only) if the true dependence is Gaussian-like. The possibility of using non-Gaussian base distribution is also discussed in [48], albeit without practical implementation. Additionally, the marginals and the vector copula in their method are learned jointly rather than separately as in our method, and they continue to treat PMI as a density ratio $p(\mathbf{x}, \mathbf{y})/p(\mathbf{x})p(\mathbf{y})$, unlike our method which treats it as a density $c(\mathbf{u}_X, \mathbf{u}_Y)$.

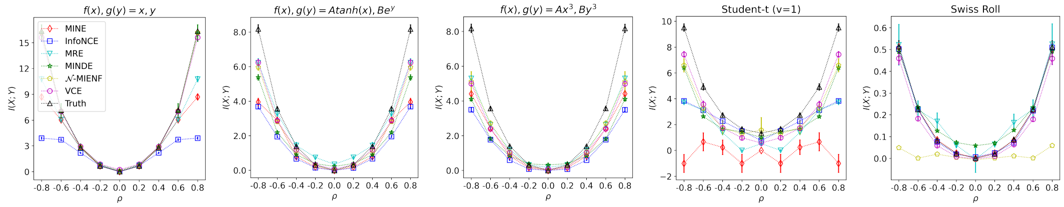
6 Experiments

Baselines. We consider five representative neural estimators in the field: MINE [20], InfoNCE [21], MRE [41], MINDE [19] and $\mathcal{N}$-MIENF [48]. The first three methods are critic-based whereas the latter three are generative model-based. MRE is chosen as the representative of state-of-the-art discriminative methods, which is specifically designed to address the high-discrepancy issue in these methods. MINDE is chosen to represent the state-of-the-art generative methods, which leverages powerful diffusion model in MI estimation. Further baselines are considered in Appendix B2.

Hyperparams. For the vector copula in VCE, we consider mixtures with $1, 4, 8, 16, 32$ components.

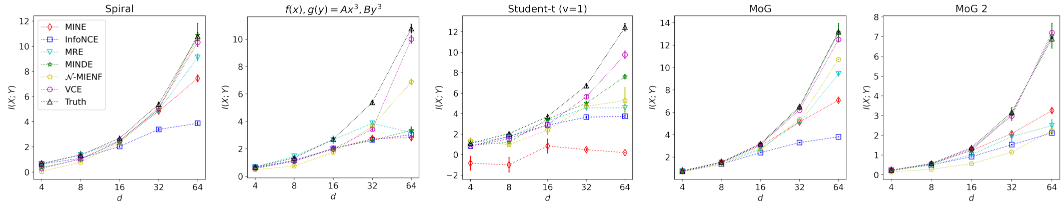
Neural architecture, optimizer and training details. Please refer to appendix B1 for more details.

In the following evaluation, we primarily focus on evaluating the VCE estimator (Algorithm 1), and present the results of the alternative VCE' estimator (Algorithm 2) in the appendix. All results are collected through 8 independent runs. Error bars reported are the standard deviations (std) of the runs.



(a) $X, Y \sim \mathcal{N}$ (b) $\mathbf{A}\tanh(X), \mathbf{B}e^{Y}$ (c) $\mathbf{A}X^3, \mathbf{B}Y^3$ (d) Student-t (e) Swiss Roll

Figure 2: Comparing MI estimators under various dependence strengths ρ. Data in cases (b)(c) are generated by first sampling $X, Y \sim \mathcal{N}$ as in case (a), then transforming them with the shown transformations. The dimensionalities of the data in the five cases are 64, 32, 32, 32, 2 respectively.



(a) Spiral (b) $\mathbf{A}X^3, \mathbf{B}Y^3$ (c) Student-t (d) MoG 1 (e) MoG 2

Figure 3: Comparing different MI estimators under various data dimensionality d and fixed dependence levels. MoG corresponds to mixture of Gaussians. Spiral corresponds to spiral transformation.

6.1 Synthetic distributions

Setups. In [58], a diverse set of models with known MI are developed to comprehensively evaluate MI estimators. We consider representative cases from this benchmark, further extending it by (a) considering varying dependence strengths for each chosen case; (b) employing mixing matrices $\mathbf{A}, \mathbf{B}$ to couple the dimensions in X and Y respectively. We also include the mixture models in [49] to enrich our tests. Together, our test cases cover non-Gaussianity, skewness, heterogeneous marginals, long tails, low-dimensional manifold structure, coupling dimensions, high-dimensionality, varying dependence strengths and non-Gaussian dependence structure. Each test case contains $n = 10^4$ data.

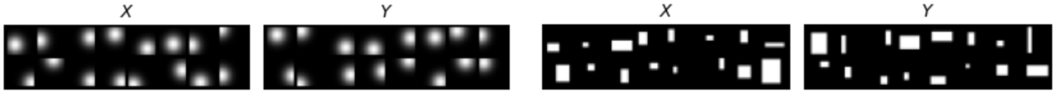
Results. Figure 2 and Figure 3 compare the performance of different MI estimators[3]. Overall, VCE provides good estimates in *all* scenarios, consistently ranking among the top performers.

Compared to discriminative methods e.g., MINE and InfoNCE, VCE demonstrates significant advantages, particularly in high MI settings (e.g. strong dependence level ρ or high dimensionality d). This advantage may be because our method avoids directly comparing two highly distinct distributions as in these methods, which is challenging. The advantage may also attribute to the better complexity-capacity trade-off in our method, which avoids an overly powerful model for the copula.

Compared to the generative method $\mathcal{N}$-MIENF, VCE demonstrates advantages in scenarios involving non-Gaussian dependence structures (see e.g. the MoG cases and 64D t-distribution). In such cases, $\mathcal{N}$-MIENF's assumption of a Gaussian dependence structure falls short in capturing the true dependence structure. This underscores the pitfalls of using a overly simplified model for the copula.

We specifically discuss two challenging cases highlighted in prior works [58, 19]: (a) Spiral transformation, which highly transforms the original data; and (b) multivariate t-distribution with degree of freedom $\nu = 1$, which exhibit heavy-tailed dependence. For these two highly challenging scenarios, VCE and MINDE are the only two methods that can simultaneously provide reasonable estimates in *both* cases, with VCE outperforming MINDE in other settings (see e.g., Figure 2.c and Figure 3.b).

[3] Comparison to classic copula-based MI estimators and further discriminative estimators is in Appendix B2.



(a) **Gaussian plates**

(b) **Rectangles**

Figure 4: The image dataset [59], which contains images of rectangles and Gaussian plates.

Method	Gaussian Plates			Rectangles		
	$I(X;Y)=1$	$I(X;Y)=3$	$I(X;Y)=7$	$I(X;Y)=1$	$I(X;Y)=3$	$I(X;Y)=7$
MINE	0.89 ± 0.07	2.86 ± 0.24	5.46 ± 0.27	0.81 ± 0.13	$\mathbf{2.57 \pm 0.26}$	5.39 ± 0.23
InfoNCE	0.86 ± 0.14	2.63 ± 0.13	3.83 ± 0.12	0.78 ± 0.17	2.49 ± 0.28	3.86 ± 0.15
MRE	1.23 ± 0.16	2.85 ± 0.21	5.91 ± 0.28	0.82 ± 0.24	2.56 ± 0.48	$\mathbf{5.45 \pm 0.31}$
$\mathcal{N}$-MIENF	0.74 ± 0.12	2.42 ± 0.16	3.85 ± 0.22	0.54 ± 0.13	0.76 ± 0.14	1.54 ± 0.11
VCE	$\mathbf{0.92 \pm 0.04}$	$\mathbf{2.93 \pm 0.12}$	$\mathbf{6.53 \pm 0.36}$	$\mathbf{0.83 \pm 0.12}$	2.27 ± 0.23	5.02 ± 0.14

Table 1: Comparing different MI estimators on the image benchmark proposed in [59].

6.2 Image dataset with known MI

Setups. We next consider the benchmark [59], which contains correlated images X and Y; see Figure 4. Here $X \in \mathbb{R}^{16 \times 16}$ and $Y \in \mathbb{R}^{16 \times 16}$, and the ground truth $I(X;Y)$ is known for this dataset. Following recent works [59, 3], we preprocess these high-dimensional image data by an autoencoder $e : \mathbb{R}^{16 \times 16} \to \mathbb{R}^{d'}$, which proves effective in reducing data dimensionality while preserving key information. The quality of such compression w.r.t d' is analyzed theoretically and empirically in Appendix A5 and B2, based on which we set $d' = 16$. A total number of $10,000$ data is used. Note that while the dependence between X and Y are Gaussian for this dataset [59], the dependence structure for the compressed data can be non-Gaussian even if the compression is near-lossless.

Results. Table 1 compares the performance of different MI estimators on this task. Our estimator consistently outperforms the recent $\mathcal{N}$-MIENF estimator on this dataset, and it shows highly competitive performance against discriminative methods. However, our method performs slightly worse than discriminative methods in the Rectangles case. One reason why our approach loses to discriminative approaches in the Rectangles case may be that the underlying dependence structure of the preprocessed data is highly complex in this case, which is difficult to model effectively with a single vector Gaussian copula or even a reasonable mixture of such copulas. Discriminative methods, on the contrary, adopt a neural network-based parameterization of the vector copula, being inherently more flexible. These results highlight the limitation of model-based parameterization of the vector copula density in certain cases. Nonetheless, our estimator still provides a highly reliable estimate.

6.3 Embeddings of language models

Setups. We further consider a real-world dataset in natural language processing. It consists of pairs of embeddings from a language model (LM) [60, 61] computed on the IMDB dataset [62], which contains negative or positive movie comments; see Table 2. The ground truth MI of this dataset is unknown, but it can be computed numerically accurately; see Appendix B1. A total number of $n = 4 \times 10^3$ data are used. Similar to the previous task, we preprocess data by an autoencoder $e : \mathbb{R}^{d_{\text{LM}}} \to \mathbb{R}^{16}$, with d_{LM} being the dimensionality of the LM's embeddings. The quality of such compression is empirically studied in Appendix B2, which is near-lossless.

Results. Table 3 summarizes the results for this dataset. In this scenario, where the underlying mutual information (MI) is relatively low, our method does not show a significant advantage over discriminative methods. This is likely because for this dataset, the high-discrepancy issue [40, 41, 29, 32] is not significant, and discriminative methods offer a more flexible parameterization of the vector copula density c than our method (see Section 5). Nonetheless, our method still provides an estimate close to discriminative methods, and it significantly outperforms the generative method $\mathcal{N}$-MIENF.

6.4 Further analysis and ablation studies

We conduct further analysis on the effect of *model selection* and *separate learning* in Appendix B2.

	X	Y
1	`(positive) I thought this was a wonderful way to spend time on ...`	`(positive) If you like original gut wrenching laughter you will like ...`
2	`(negative) So im not a big fan of Boll's work but then ...`	`(positive) This a fantastic movie of three prisoners who become famous...`

Table 2: The text benchmark, which contains reviews of positive or negative movie comments.

Method	$I(X;Y) \approx 2.1$	$I(X;Y) \approx 0.9$
MINE	1.83 ± 0.04	0.71 ± 0.05
InfoNCE	1.64 ± 0.09	0.70 ± 0.06
MRE	1.72 ± 0.07	1.23 ± 0.02
$\mathcal{N}$-MIENF	0.91 ± 0.05	0.43 ± 0.03
VCE	$\mathbf{2.01 \pm 0.04}$	$\mathbf{0.83 \pm 0.01}$

(a) Llama-3 13B

Method	$I(X;Y) \approx 1.5$	$I(X;Y) \approx 0.2$
MINE	$\mathbf{1.42 \pm 0.04}$	0.18 ± 0.02
InfoNCE	1.41 ± 0.03	0.19 ± 0.04
MRE	1.23 ± 0.09	0.31 ± 0.09
$\mathcal{N}$-MIENF	0.73 ± 0.03	0.11 ± 0.02
VCE	1.22 ± 0.02	$\mathbf{0.19 \pm 0.02}$

(b) BERT

Table 3: Comparing different MI estimators on the text dataset. Left: evaluation on the embeddings of Llama-3 13B model [61]. Right: evaluation on the embeddings of a BERT model [60].

7 Conclusion

In this work, we introduced a new mutual information (MI) estimator grounded in recent vector copula theory. A fundamental difference to existing approaches is the explicit disentanglement of marginal distributions and dependence structure in our method. This separation enables more flexible and fine-grained modeling, avoiding the pitfalls of both overly simplistic or excessively complex approaches, and reducing overall learning difficulty via strategic factorization of the original estimation problem. Extensive experiments demonstrate our method's effectiveness and robustness.

Beyond the development of practical estimator, our research also offers fresh perspectives on MI estimation. By viewing PMI as a density rather than a density ratio, we open new avenues for modeling. Additionally, our approach to vector rank computation generalizes the classical copula transformation and holds promise as a versatile preprocessing step for a broad range of MI estimators. Finally, by reinterpreting existing estimators through the lens of vector copula theory, we obtain new insights into the parameterization and learning of different estimators and the underlying trade-offs.

Copulas have been widely used for MI estimate [63, 55, 64, 56, 65, 33, 51, 45, 66]. Existing methods primarily focus on *classic copulas*, where the copula transformation is applied independently to each univariate marginal to better account for the marginal-invariant property of MI. This strategy has been shown to improve accuracy and reduce variance [56, 65]. We go one step further by using *vector copulas*, where the transformation jointly considers all dimensions of the multivariate marginals. This can be seen as a generalization of classic copula transformation, where we not only consider MI's invariance to *element-wise* bijections but also to *any* diffeomorphisms. Another key difference lies in that these works still treat PMI as a density ratio, whereas our work treats PMI as a density.

We note that, while powerful, our estimator is not a panacea. One limitation of our method is that it relies on the two marginal distributions to be reasonably modeled. While marginal distributions are far easier to learn than the joint distribution, they can still be challenging to learn for high-dimensional data e.g., images. Fortunately, dimensionality reduction techniques [3, 13] help to mitigate this issue. Another limitation lies in the flexibility of our model-based parameterization of vector copula, which can be less flexible than neural network methods. However, as our method strikes a good trade-off between complexity and capacity across diverse cases, we consider it as a highly competitive method.

Acknowledgments

AW acknowledges supports from a Turing AI Fellowship under grant EP/V025279/1, the Alan Turing Institute, and the Leverhulme Trust via the Leverhulme Centre for the Future of Intelligence. MUG was supported in part by Gen AI - the AI Hub for Generative Models, funded by EPSRC (EP/Y028805/1). YC acknowledge supports from the Cambridge Trust and the Qualcomm Innovation Fellowship. ZO acknowledges supports from Lee Family Scholarship via Imperial College London.

References

[1] Justin B Kinney and Gurinder S Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, 111(9):3354–3359, 2014.

[2] Tm Cover, Ja Thomas, and J Wiley. *Elements of information theory.* Tsinghua University Pres, 2003.

[3] Gokul Gowri, Xiao-Kang Lun, Allon M Klein, and Peng Yin. Approximating mutual information of high-dimensional variables using learned representations. *arXiv preprint arXiv:2409.02732*, 2024.

[4] Antonio Reverter and Eva KF Chan. Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks. *Bioinformatics*, 24(21):2491–2497, 2008.

[5] Lin Song, Peter Langfelder, and Steve Horvath. Comparison of co-expression measures: mutual information, correlation, and model based indices. *BMC bioinformatics*, 13:1–21, 2012.

[6] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[7] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[8] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2019.

[9] Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. On mutual information in contrastive learning for visual representations. *arXiv preprint arXiv:2005.13149*, 2020.

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 29, 2016.

[11] Yanzhi Chen, Michael U Gutmann, and Adrian Weller. Is learning summary statistics necessary for likelihood-free inference? In *International Conference on Machine Learning*, pages 4529–4544. PMLR, 2023.

[12] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[13] Yanzhi Chen, Dinghuai Zhang, Michael Gutmann, Aaron Courville, and Zhanxing Zhu. Neural approximate sufficient statistics for implicit models. *arXiv preprint arXiv:2010.10079*, 2020.

[14] Zexuan Qiu, Qinliang Su, Zijing Ou, Jianxing Yu, and Changyou Chen. Unsupervised hashing with contrastive information bottleneck. *arXiv preprint arXiv:2105.06138*, 2021.

[15] Yanzhi Chen, Yingzhen Li, Adrian Weller, et al. Scalable infomin learning. *Advances in Neural Information Processing Systems*, 35:2226–2239, 2022.

[16] Sayedeh Leila Noorbakhsh, Binghui Zhang, Yuan Hong, and Binghui Wang. {Inf2Guard}: An {Information-Theoretic} framework for learning {Privacy-Preserving} representations against inference attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2405–2422, 2024.

[17] Bao Duong and Thin Nguyen. Diffeomorphic information neural estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7468–7475, 2023.

[18] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*, pages 1779–1788. PMLR, 2020.

[19] Giulio Franzese, Mustapha BOUNOUA, and Pietro Michiardi. Minde: Mutual information neural diffusion estimation. In *The Twelfth International Conference on Learning Representations*, 2023.

[20] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[21] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[22] Qing Guo, Junya Chen, Dong Wang, Yuewei Yang, Xinwei Deng, Jing Huang, Larry Carin, Fan Li, and Chenyang Tao. Tight mutual information estimation with contrastive fenchel-legendre optimization. *Advances in Neural Information Processing Systems*, 35:28319–28334, 2022.

[23] Nunzio Alexandro Letizia, Nicola Novello, and Andrea M Tonello. Mutual information estimation via f-divergence and data derangements. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[24] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[25] Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

[26] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.

[27] Yihong Wu and Pengkun Yang. Minimax rates of entropy estimation on large alphabets via best polynomial approximation. *IEEE Transactions on Information Theory*, 62(6):3702–3720, 2016.

[28] Andrew R. Barron and Kai Guo. Convergence of smoothed empirical measures with applications to entropy estimation. *IEEE Transactions on Information Theory*, 68(8):5326–5353, 2022.

[29] David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, pages 875–884. PMLR, 2020.

[30] Yanqin Fan and Marc Henry. Vector copulas. *Journal of Econometrics*, 234(1):128–150, 2023.

[31] M Sklar. Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, volume 8, pages 229–231, 1959.

[32] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*, 2019.

[33] Shashank Singh and Barnabás Póczos. Nonparanormal information estimation. In *International Conference on Machine Learning*, pages 3210–3219. PMLR, 2017.

[34] Yanzhi Chen and Michael U Gutmann. Adaptive gaussian copula abc. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1584–1592. PMLR, 2019.

[35] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

[36] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *arXiv preprint arXiv:1906.04032*, 2019.

[37] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[38] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

[39] Ruizhi Liao, Daniel Moyer, Polina Golland, and William M Wells. Demi: Discriminative estimator of mutual information. *arXiv preprint arXiv:2010.01766*, 2020.

[40] Benjamin Rhodes, Kai Xu, and Michael U Gutmann. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020.

[41] Akash Srivastava, Seungwook Han, Kai Xu, Benjamin Rhodes, and Michael U Gutmann. Estimating the density ratio between distributions with high discrepancy using multinomial logistic regression. *Transactions on Machine Learning Research*, 2023(3):1–23, 2023.

[42] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on pure and applied mathematics*, 36(2):183–212, 1983.

[43] Victor Chernozhukov, Alfred Galichon, Marc Hallin, and Marc Henry. Monge-kantorovich depth, quantiles, ranks and signs. *Annals of Statistics*, 45(1):223–256, 2017.

[44] Manuel Davy and Arnaud Doucet. Copulas: a new insight into positive time-frequency distributions. *IEEE signal processing letters*, 10(7):215–218, 2003.

[45] Jian Ma and Zengqi Sun. Mutual information is copula entropy. *Tsinghua Science & Technology*, 16(1):51–54, 2011.

[46] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540. PMLR, 2018.

[47] Zhengyang Hu, Song Kang, Qunsong Zeng, Kaibin Huang, and Yanchao Yang. Infonet: Neural estimation of mutual information without test-time optimization. In *Forty-first International Conference on Machine Learning*.

[48] Ivan Butakov, Aleksandr Tolmachev, Sofia Malanchuk, Anna Neopryatnaya, and Alexey Frolov. Mutual information estimation via normalizing flows. *Advances in Neural Information Processing Systems*, 37:3027–3057, 2024.

[49] Paweł Czyż, Frederic Grabowski, Julia E Vogt, Niko Beerenwinkel, and Alexander Marx. The mixtures and the neural critics: On the pointwise mutual information profiles of fine distributions. *arXiv preprint arXiv:2310.10240*, 2023.

[50] Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. Tactis-2: Better, faster, simpler attentional copulas for multivariate time series. *arXiv preprint arXiv:2310.01327*, 2023.

[51] David Huk, Mark Steel, and Ritabrata Dutta. Your copula is a classifier in disguise: classification-based copula density estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 3790–3798. PMLR, 2025.

[52] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[53] Sudipto Mukherjee, Himanshu Asnani, and Sreeram Kannan. Ccmi: Classifier based conditional mutual information estimation. In *Uncertainty in artificial intelligence*, pages 1083–1093. PMLR, 2020.

[54] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[55] Houman Safaai, Arno Onken, Christopher D Harvey, and Stefano Panzeri. Information estimation using nonparametric copulas. *Physical Review E*, 98(5):053302, 2018.

[56] Xianli Zeng, Yingcun Xia, and Howell Tong. Jackknife approach to the estimation of mutual information. *Proceedings of the National Academy of Sciences*, 115(40):9956–9961, 2018.

[57] Soumik Purkayastha and Peter X-K Song. fastmi: A fast and consistent copula-based nonparametric estimator of mutual information. *Journal of Multivariate Analysis*, 201:105270, 2024.

[58] Paweł Czyż, Frederic Grabowski, Julia E Vogt, Niko Beerenwinkel, and Alexander Marx. Beyond normal: On the evaluation of mutual information estimators. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[59] Ivan Butakov, Alexander Tolmachev, Sofia Malanchuk, Anna Neopryatnaya, Alexey Frolov, and Kirill Andreev. Information bottleneck analysis of deep neural networks via lossy compression. In *The Twelfth International Conference on Learning Representations*, 2024.

[60] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[61] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[62] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[63] Yves-Laurent Kom Samo. Inductive mutual information estimation: A convex maximum-entropy copula approach. In *International Conference on Artificial Intelligence and Statistics*, pages 2242–2250. PMLR, 2021.

[64] Amor Keziou and Philippe Regnault. Semiparametric estimation of mutual information and related criteria: Optimal test of independence. *IEEE Transactions on Information Theory*, 63(1):57–71, 2016.

[65] Nunzio A Letizia, Nicola Novello, and Andrea M Tonello. Copula density neural estimation. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[66] Gery Geenens. Towards a universal representation of statistical dependence. *arXiv preprint arXiv:2302.08151*, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] ,

   Justification: The abstract and introduction directly reflects the exact contribution of the work.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] ,

   Justification: Limitation is explicitly discussed in conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes] ,

Justification: All proofs have a clear declaration of assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes] ,

   Justification: Code is available in the provided repo, and the details of experiments and implementations are also documented in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes] ,

Justification: source code is provided through the github link provided.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes] ,

Justification: all experimental details are provided in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars (which correspond to the standard errors) are reported, which are collected from 8 independent runs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] ,

   Justification: please see the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes] ,

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes] ,

    Justification: the work is a general method and its societal impacts have been discussed in introduction and conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: no such risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: all code and data are taken from public available resource without violating any license and terms.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA] .

    Justification: no new assets introduced at this stage.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLM is only used for proofreading.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.