Estimating Total Correlation with Mutual Information Estimators

Ke Bai[*]
Duke University

Pengyu Cheng[*]
Tencent AI Lab

Weituo Hao
ByteDance

Ricardo Henao
KAUST

Lawrence Carin
KAUST

Abstract

Total correlation (TC) is a fundamental concept in information theory which measures statistical dependency among multiple random variables. Recently, TC has shown noticeable effectiveness as a regularizer in many learning tasks, where the correlation among multiple latent embeddings requires to be jointly minimized or maximized. However, calculating precise TC values is challenging, especially when the closed-form distributions of embedding variables are unknown. In this paper, we introduce a unified framework to estimate total correlation values with sample-based mutual information (MI) estimators. More specifically, we discover a relation between TC and MI and propose two types of calculation paths (tree-like and line-like) to decompose TC into MI terms. With each MI term being bounded, the TC values can be successfully estimated. Further, we provide theoretical analyses concerning the statistical consistency of the proposed TC estimators. Experiments are presented on both synthetic and real-world scenarios, where our TC estimators demonstrate effectiveness in all TC estimation, minimization, and maximization tasks. The code is available at *https://github.com/Linear95/TC-estimation*.

1 INTRODUCTION

Statistical dependency measures the association (correlation) of variables or factors in models and systems, and constitutes one of the key considerations in various scientific domains including statistics (Granger and Lin, 1994; Jiang et al., 2015), robotics (Julian et al., 2014; Charrow et al., 2015), bioinformatics (Lachmann et al., 2016; Zea et al., 2016), and machine learning (Chen et al., 2016; Alemi et al., 2016; Hjelm et al., 2018). In deep learning, statistical dependency has been applied as learning objective or regularizer in many well-known training frameworks, such as information bottleneck (Alemi et al., 2016), disentangled representation learning (Chen et al., 2018; Peng et al., 2019; Cheng and Li, 2022) and contrastive learning (Chen et al., 2020; Gao et al., 2021). Recent neural network studies have also demonstrated the benefits of considering statistical dependency in terms of model robustness (Zhu et al., 2020), fairness (Creager et al., 2019), interpretability (Chen et al., 2016; Cheng et al., 2020b), *etc.*

Among diverse measurement approaches for statistical dependency, the concept of mutual information (MI) is one of the most commonly used, especially in deep model training (Alemi et al., 2016; Belghazi et al., 2018; Chen et al., 2020). Given two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$ with joint and marginal distributions $p(\boldsymbol{x}, \boldsymbol{y})$, $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$, respectively, their mutual information is defined as:

$$\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}\Big[\log \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}\Big]. \tag{1}$$

Recently, mutual information has been used as a training criterion to deliver noticeable performance improvement for deep models on various learning tasks such as conditional generation (Chen et al., 2016; Cheng et al., 2020b), domain adaptation (Gholami et al., 2020; Cheng et al., 2020a), representation learning (Chen et al., 2020; Gao et al., 2021; Yuan et al., 2021), and model debiasing (Song et al., 2019; Cheng et al., 2021). However, standard MI in equation 1 only handles the statistical dependency between two variables. When considering correlation among multiple variables, MI requires computation between each pair of variable, which leads to a quadratic scaling in computation cost. To address this problem, total correlation (TC) (Watanabe, 1960) or multi-information (Studenỳ and Vejnarová, 1998) has been proposed for multi-variable scenarios:

$$\begin{aligned}\mathcal{TC}(\boldsymbol{X}) =& \mathcal{TC}(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n) \\ =& \mathbb{E}_{p(\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_n)}\Big[\log \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)}{p(\boldsymbol{x}_1)p(\boldsymbol{x}_2)\dots p(\boldsymbol{x}_n)}\Big].\end{aligned} \tag{2}$$

TC has been also proven effective to enhance machine learning models in many tasks such as independent component analysis (Cardoso, 2003), structure discovery (Ver Steeg and Galstyan, 2014) and disentangled representation learning (Chen et al., 2018; Locatello et al., 2019a; Kim and Mnih, 2018). However, TC suffers from the same practical

[*]Equal contribution. Corresponding to Pengyu Cheng.

Proceedings of the 26$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

problem as MI, namely, that the exact values of TC are difficult to calculate without the availability of the closed-form distributions, or only relying on samples either via the kernel density estimation (KDE) (Kandasamy et al., 2015; Singh and Póczos, 2014) and k-Nearest Neighbor (k-NN) (Pál et al., 2010; Kraskov et al., 2004a; Gao et al., 2018), which may perform well on data with low dimensionality but failed on the high one. Furthermore, previous works on disentangled representation learning (Chen et al., 2018; Gao et al., 2019) strongly assume that both the joint distribution $p(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)$ and marginal distributions $\{p(\boldsymbol{x}_i)\}_{i=1}^n$ follow Gaussian distributions, so that the TC value can be calculated in closed-form. Poole et al. (2019) proposed an upper bound of TC by further introducing an auxiliary variable $\boldsymbol{y}$. With a strong assumption that given $\boldsymbol{y}$ and $\boldsymbol{x}_i|\boldsymbol{y}$, all variables are conditionally independent, $p(\boldsymbol{X}|\boldsymbol{y}) = \prod_{i=1}^n p(\boldsymbol{x}_i|\boldsymbol{y})$, Poole et al. (2019) showed that $\mathcal{TC}(\boldsymbol{X}) = \sum_{i=1}^n \mathcal{I}(\boldsymbol{x}_i;\boldsymbol{y}) - \mathcal{I}(\boldsymbol{X};\boldsymbol{y})$, *i.e.*, that the TC value can be bounded by MI estimators (Belghazi et al., 2018; Poole et al., 2019; Cheng et al., 2020a). All these TC estimation methods require additional strong assumptions on the distributions of data samples, which heavily limits their application in practice.

In this paper, we propose a new **T**otal **C**orrelation **E**stimator via **L**inear **D**ecomposition (TCeld), which importantly, does not require any assumptions about sample distributions. More specifically, we discover a linear decomposition relation between TC and mutual information (MI). Based on this relationship, we linearly split the TC into several MI terms, then estimate each MI term aided with variational MI estimators. With two different TC decomposition paths (*line-like* and *tree-like*), we obtain two types of TC estimators, $\widehat{\mathcal{TC}}_{\text{Tree}}$ and $\widehat{\mathcal{TC}}_{\text{Line}}$, respectively. Moreover, we prove that the nice properties of MI estimators, such as consistency, are maintained in the corresponding TC estimators, thanks to the linearity of the proposed TC decomposition. In the experiments, we first test the estimation quality of our TC estimators on simulation data sampled from multi-variate Gaussian distributions, then apply the TC estimators to two real-world TC optimization tasks. The numerical results demonstrate the effectiveness of the proposed TC estimators under both TC estimation and optimization scenarios.

2 BACKGROUND

2.1 Sample-based Mutual Information Estimators

Although mutual information (MI) is a fundamental tool for measuring statistical dependency between two variables, calculating MI values with only samples provided is challenging, especially when the closed-form distributions of variables are unknown. To estimate MI values with samples, several variational MI estimators have been introduced. Barber and Agakov (2003) approximate the conditional distribution $p(\boldsymbol{x}|\boldsymbol{y})$ between $\boldsymbol{x}$ and $\boldsymbol{y}$ by a variational distribution $q(\boldsymbol{x}|\boldsymbol{y})$, and derive:

$$\mathcal{I}_{\text{BA}} := \text{H}(\boldsymbol{x}) + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\log q(\boldsymbol{x}|\boldsymbol{y})], \tag{3}$$

with $\text{H}(\boldsymbol{x})$ as the entropy of $\boldsymbol{x}$. Utilizing the Donsker-Varadhan representation (Donsker and Varadhan, 1983) of KL-divergence, Belghazi et al. (2018) obtain an MI Neural Estimator (MINE) with a score network $\phi(\cdot,\cdot)$:

$$\begin{aligned}\mathcal{I}_{\text{MINE}} :=& \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\phi(\boldsymbol{x},\boldsymbol{y})] \\ &- \log(\mathbb{E}_{p(\boldsymbol{x})p(\boldsymbol{y})}[\exp(\phi(\boldsymbol{x},\boldsymbol{y}))].\end{aligned} \tag{4}$$

Nguyen, Wainwright, and Jordan (NWJ) derive another lower bound considering MI as in a f-divergence form Nguyen et al. (2010), which also requires a score network $\phi(\cdot,\cdot)$:

$$\mathcal{I}_{\text{NWJ}} := \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\phi(\boldsymbol{x},\boldsymbol{y})] - \mathbb{E}_{p(\boldsymbol{x})p(\boldsymbol{y})}[e^{\phi(\boldsymbol{x},\boldsymbol{y})-1}]. \tag{5}$$

With Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2010), Oord et al. (2018) propose a MI lower bound called InfoNCE, based on a group of samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$, to obtain a low-variance estimator:

$$\mathcal{I}_{\text{InfoNCE}} := \mathbb{E}[\frac{1}{N}\sum_{i=1}^N \log \frac{\exp(\phi(\boldsymbol{x}_i,\boldsymbol{y}_i))}{\frac{1}{N}\sum_{j=1}^N \exp(\phi(\boldsymbol{x}_i,\boldsymbol{y}_j))}]. \tag{6}$$

Different from the MI lower bounds introduced above, Cheng et al. (2020a) derive a Contrastive Log-ratio Upper Bound (CLUB), which is also based on a variational approximation $q(\boldsymbol{x}|\boldsymbol{y})$ of $p(\boldsymbol{x}|\boldsymbol{y})$:

$$\mathcal{I}_{\text{CLUB}} := \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\log q(\boldsymbol{x}|\boldsymbol{y})] - \mathbb{E}_{p(\boldsymbol{x})p(\boldsymbol{y})}[\log q(\boldsymbol{x}|\boldsymbol{y})] \tag{7}$$

2.2 Statistical Properties of Estimators

Given observed samples $\boldsymbol{x}^1, \boldsymbol{x}^2, \dots \boldsymbol{x}^m \sim p(\boldsymbol{x})$, a statistic is defined as $T(\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^m)$, where $T(\cdot)$ is an arbitrary real-valued function taking samples $\boldsymbol{x}^1, \boldsymbol{x}^2, \dots \boldsymbol{x}^m$ as inputs (DeGroot and Schervish, 2012). Sample-based estimators to calculate statistical dependency, *i.e.* mutual information and total correlation, are also examples of statistics. To evaluate the performance of a statistic, *statistical properties* are introduced to describe behaviors of the statistic under different data situations (DeGroot and Schervish, 2012) (*e.g.*, with large/small sample number, with/without outlier). For estimation of mutual information and total correlation, we follow prior works (Paninski, 2003; Belghazi et al., 2018) and will mainly consider the following three key properties:

Definition 2.1 (Unbiasedness). *An estimator $\hat{R} = T(\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^m)$ is unbiased of the ground true value R_p with respect to the distribution $p(\boldsymbol{x})$, if $\mathbb{E}_{p(\boldsymbol{x})}[\hat{R}] = R_p$.*

Definition 2.2 (Consistency). *An estimator $\hat{R}_m = T(\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^m)$ is consistent to the ground true value R_p with respect to the distribution $p(\boldsymbol{x})$, if $\forall \varepsilon > 0$, $\lim_{m\to\infty} \mathbb{P}\{|\hat{R}_m - R_p| \geq \varepsilon\} = 0$.*

Definition 2.3 (Strong Consistency)**.** *An estimator $\hat{R}_m = T(\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^m)$ is strongly consistent to the ground true value R_p with respect to the distribution $p(\boldsymbol{x})$, if $\forall \varepsilon > 0$, there exists an integer $M > 0$ such that $\forall m > M$, $|\hat{R}_m - R_p| \le \varepsilon$ almost surely.*

3 TOTAL CORRELATION ESTIMATION

Suppose m groups of data samples $\{\boldsymbol{X}^j\}_{j=1}^m = \{(\boldsymbol{x}_1^j, \boldsymbol{x}_2^j, \dots, \boldsymbol{x}_n^j)\}_{j=1}^m$ are observed from an unknown distribution $p(\boldsymbol{X}) = p(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)$. We seek to estimate the total correlation (TC) of these n variables as in equation 2. Below we first describe the proposed sample-based TC estimators, then analyze their statistical properties.

3.1 Sample-based TC Estimators

With the definition of total correlation (TC) and mutual information (MI) in equation 2 and equation 1, we discover a connection between TC and MI and summarize it in the following Theorem 3.1.

Theorem 3.1. *Let $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)$ be a group of random variables. Suppose set $\mathcal{A} = \{i_1, i_2, \dots, i_k\} \subseteq \{1, 2, \dots, n\}$ is an index subgroup. $\bar{\mathcal{A}} = \{i : i \notin \mathcal{A}\}$ is the complementary set of $\mathcal{A}$. Denote $\boldsymbol{X}_{\mathcal{A}} = (\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \dots, \boldsymbol{x}_{i_k})$ as the selected variables from $\boldsymbol{X}$ with the indexes $\mathcal{A}$. Then we have $\mathcal{TC}(\boldsymbol{X}) = \mathcal{TC}(\boldsymbol{X}_{\mathcal{A}}) + \mathcal{TC}(\boldsymbol{X}_{\bar{\mathcal{A}}}) + \mathcal{I}(\boldsymbol{X}_{\mathcal{A}}; \boldsymbol{X}_{\bar{\mathcal{A}}})$.*

Theorem 3.1 underscores the insight that the TC of a group of variables $\boldsymbol{X}$ can be decomposed into the TC of two subgroups $\mathcal{TC}(\boldsymbol{X}_{\mathcal{A}})$ and $\mathcal{TC}(\boldsymbol{X}_{\bar{\mathcal{A}}})$ and the MI between the two subgroups $\mathcal{I}(\boldsymbol{X}_{\mathcal{A}}; \boldsymbol{X}_{\bar{\mathcal{A}}})$. Therefore, we can recursively represent the TC of the subgroups in terms of MI terms for lower-level subgroups. With this decomposition strategy, we can effectively cast $\mathcal{TC}(\boldsymbol{X})$ into a summation of MI terms between variable subgroups. Note that the decomposition form depends on the separation of variables in each subgroup. In the following, we introduce two types of variable separation strategies: *line-like* and *tree-like* decomposition as shown in Figure 1.

Line-like Decomposition In each variable subgroup separation, our line-like decomposition strategy splits out a single variable. Let $\boldsymbol{X}_{i:j} = (\boldsymbol{x}_i, \boldsymbol{x}_{i+1}, \dots, \boldsymbol{x}_j)$ denote a subset of variables with indexes from i to j. Then we can extend Theorem 3.1 to the following Corollary 3.1.1 and Corollary 3.1.2. Based on Corollary 3.1.1, the line-like decomposition can be dynamically described as:

$$\mathcal{TC}(\boldsymbol{X}_{1:i+1}) = \mathcal{TC}(\boldsymbol{X}_{1:i}) + \mathcal{I}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1}). \tag{8}$$

Iteratively applying equation 8 to the remaining TC term, we derive the representation of $\mathcal{TC}(\boldsymbol{X})$ as the summation of MI terms in Corollary 3.1.2. With a given MI estimation method $\hat{\mathcal{I}}$ applied to each MI term in Corollary 3.1.2, our line-like TC estimator can be calculated as:

$$\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}](\boldsymbol{X}) = \sum_{i=1}^{n-1} \hat{\mathcal{I}}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1}). \tag{9}$$

Corollary 3.1.1. *Given the group $\boldsymbol{X}$ and another variable $\boldsymbol{y}$, $\mathcal{TC}(\boldsymbol{X} \cup \{\boldsymbol{y}\}) = \mathcal{TC}(\boldsymbol{X}) + \mathcal{I}(\boldsymbol{X}; \boldsymbol{y})$.*

Corollary 3.1.2. *Given $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)$, we have $\mathcal{TC}(\boldsymbol{X}) = \sum_{i=1}^{n-1} \mathcal{I}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1})$.*

Tree-like Decomposition In each variable subgroup separation, the tree-like decomposition strategy separates variables $\boldsymbol{X}_{i:j}$ into balanced variable subgroups with similar sizes in the following way:

$$\begin{aligned}\mathcal{TC}(\boldsymbol{X}_{i:j}) =& \mathcal{TC}(\boldsymbol{X}_{i:\lfloor (i+j)/2 \rfloor}) + \mathcal{TC}(\boldsymbol{X}_{\lfloor (i+j)/2 \rfloor + 1:j}) \\ &+ \mathcal{I}(\boldsymbol{X}_{i:\lfloor (i+j)/2 \rfloor}; \boldsymbol{X}_{\lfloor (i+j)/2 \rfloor + 1:j}),\end{aligned} \tag{10}$$

where $\lfloor t \rfloor$ indicates the largest integer smaller than t. In accordance with the line-like decomposition, iteratively applying this dichotomous dynamic will finally convert $\mathcal{TC}(\boldsymbol{X})$ into the summation of MI terms. Since the closed-form of this tree-like TC estimator is hard to summarize in an equation, we describe it recursively in Algorithm 1.

We call this novel TC estimator as **T**otal **C**orrelation **E**stimation with **L**inear **D**ecomposition (TCeld). From the linearity of the above two decomposition strategies, one can easily derive:

Theorem 3.2. *If an MI estimator $\hat{\mathcal{I}}$ is an MI upper (or lower) bound, then the corresponding $\widehat{\mathcal{TC}}_{Line}[\hat{\mathcal{I}}]$ and $\widehat{\mathcal{TC}}_{Tree}[\hat{\mathcal{I}}]$ are both the TC upper (or lower) bounds.*

Therefore, by selecting an MI lower bound $\hat{\mathcal{I}}_0$ and an MI upper bound $\hat{\mathcal{I}}_1$, we can limit the ground-truth TC value in a certain range, $\widehat{\mathcal{TC}}^*[\hat{\mathcal{I}}_0](\boldsymbol{X}) \le \mathcal{TC}(\boldsymbol{X}) \le \widehat{\mathcal{TC}}^*[\hat{\mathcal{I}}_1](\boldsymbol{X})$, where $\widehat{\mathcal{TC}}^*$ can be either line-like $\widehat{\mathcal{TC}}_{\text{Line}}$ or tree-like $\widehat{\mathcal{TC}}_{\text{Tree}}$.

Both the line-like and tree-like TC estimators have no additional requirement on the selection of MI estimators. However, the statistical performance of the proposed TC estimators highly depends on the selected MI estimators. To further analyze the influence of MI bounds choice for TC estimators, we discuss the statistical properties between the TC estimators and MI estimators in the next subsection.

3.2 Statistical Properties of TC Estimators

As introduced in Section 2.2, TC estimators can be also regarded as a type of statistics based on observed groups of sample data. Therefore, the aforementioned statistical properties (unbiasedness, consistency, and strong consistency in Section 2.2) are applicable to TC estimators. Thanks to the form of linear combinations of MI terms in our TC estimators, we find the following relations between TC and MI estimators in terms of statistical properties:

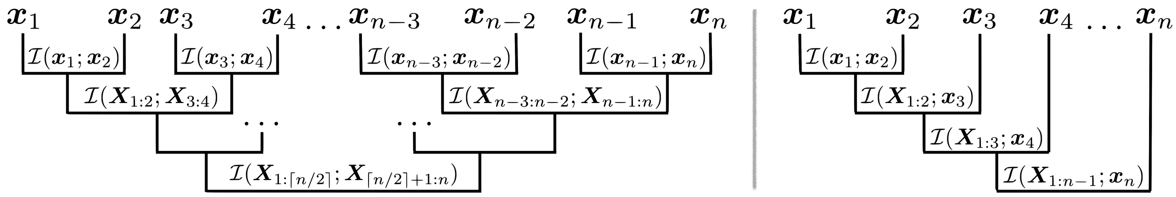

$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$\boldsymbol{x}_3$
$\boldsymbol{x}_4 \dots$
$\boldsymbol{x}_{n-3}$
$\boldsymbol{x}_{n-2}$
$\boldsymbol{x}_{n-1}$
$\boldsymbol{x}_n$
$\mathcal{I}(\boldsymbol{x}_1;\boldsymbol{x}_2)$
$\mathcal{I}(\boldsymbol{x}_3;\boldsymbol{x}_4)$
$\mathcal{I}(\boldsymbol{x}_{n-3};\boldsymbol{x}_{n-2})$
$\mathcal{I}(\boldsymbol{x}_{n-1};\boldsymbol{x}_n)$
$\mathcal{I}(\boldsymbol{X}_{1:2};\boldsymbol{X}_{3:4})$
$\mathcal{I}(\boldsymbol{X}_{n-3:n-2};\boldsymbol{X}_{n-1:n})$
$\dots$
$\dots$
$\mathcal{I}(\boldsymbol{X}_{1:\lceil n/2\rceil};\boldsymbol{X}_{\lceil n/2\rceil+1:n})$
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$\boldsymbol{x}_3$
$\boldsymbol{x}_4 \dots$
$\boldsymbol{x}_n$
$\mathcal{I}(\boldsymbol{x}_1;\boldsymbol{x}_2)$
$\mathcal{I}(\boldsymbol{X}_{1:2};\boldsymbol{x}_3)$
$\mathcal{I}(\boldsymbol{X}_{1:3};\boldsymbol{x}_4)$
$\mathcal{I}(\boldsymbol{X}_{1:n-1};\boldsymbol{x}_n)$

Figure 1: Two decomposition paths of total correlation $\mathcal{TC}(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)$. **Left** (tree-like decomposition): Divide the variables in a group into two subgroups with similar sizes. Calculate the MI between the subgroups and recursively calculate the TC of both subgroups. $\lceil n/2 \rceil$ is the smallest number larger than $n/2$. **Right** (line-like decomposition): Calculate the MI between the current group of variables and the next variable, and then add the next variable into current group.

Algorithm 1 Tree-like TC estimator $\widehat{\mathcal{TC}}_{\text{Tree}}[\hat{\mathcal{I}}]$ calculation.

```
**Prerequisite:** MI estimation method $\hat{\mathcal{I}}$, samples $\{\boldsymbol{X}^j_{1:n}\}_{j=1}^m = \{(\boldsymbol{x}^j_1, \boldsymbol{x}^j_2, \dots, \boldsymbol{x}^j_n)\}_{j=1}^m$
**Function** $\widehat{\mathcal{TC}}_{\text{Tree}}[\hat{\mathcal{I}}](\boldsymbol{X}_{i:j})$**:**
**if** $j - i < 1$ **then**
    **return** $0$
**else**
    $m = \lfloor (i+j)/2 \rfloor$
    **return** $\widehat{\mathcal{TC}}_{\text{Tree}}[\hat{\mathcal{I}}](\boldsymbol{X}_{i:m}) + \widehat{\mathcal{TC}}_{\text{Tree}}[\hat{\mathcal{I}}](\boldsymbol{X}_{m+1:j}) + \hat{\mathcal{I}}(\boldsymbol{X}_{i:m}; \boldsymbol{X}_{m+1:j})$
**end if**
```

Theorem 3.3. *If an MI estimator $\hat{\mathcal{I}}$ is unbiased, then the corresponding TC estimators $\widehat{\mathcal{TC}}_{\textit{Line}}[\hat{\mathcal{I}}]$ and $\widehat{\mathcal{TC}}_{\textit{Tree}}[\hat{\mathcal{I}}]$ are both unbiased.*

Theorem 3.4. *If an MI estimator $\hat{\mathcal{I}}$ is (strongly) consistent, then the corresponding TC estimators $\widehat{\mathcal{TC}}_{\textit{Line}}[\hat{\mathcal{I}}]$ and $\widehat{\mathcal{TC}}_{\textit{Tree}}[\hat{\mathcal{I}}]$ are both (strongly) consistent.*

The definitions of *unbiasedness* and *consistency* are introduced in Section 2.2. The details of the proofs for both theorems are shown in the Supplementary Material. These two Theorems indicate that the unbiasedness and the consistency of the MI estimators is conveniently inherited by the corresponding TC estimators. Note that Belghazi et al. (2018) have shown that MINE (in equation 4) MI estimator is strongly consistent. Consequently, we have: n

Corollary 3.4.1. *There exists an score network function family $\{\phi(\cdot,\cdot)\}$ (as described in equation 4), such that $\widehat{\mathcal{TC}}_{\textit{Line}}[\hat{\mathcal{I}}_{\textit{MINE}}]$ and $\widehat{\mathcal{TC}}_{\textit{Tree}}[\hat{\mathcal{I}}_{\textit{MINE}}]$ are strongly consistent.*

Apart from the MINE MI estimator, we also analyze the asymptotic behaviors of other MI variational estimators (InfoNCE in equation 6, NWJ in equation 5, and CLUB in equation 7). We found that both InfoNCE and NWJ are strongly consistent while CLUB does not guarantee the consistency (supportive proofs are provided in Supplementary Material). Therefore, we summarize the consistency of the corresponding TC estimators in the corollaries below:

Corollary 3.4.2. *For any MI estimator $\hat{\mathcal{I}} \in \{\hat{\mathcal{I}}_{\textit{InfoNCE}}, \hat{\mathcal{I}}_{\textit{NWJ}}\}$, there exists a score network function family $\{\phi(\cdot,\cdot)\}$, such that $\widehat{\mathcal{TC}}_{\textit{Line}}[\hat{\mathcal{I}}]$ and $\widehat{\mathcal{TC}}_{\textit{Tree}}[\hat{\mathcal{I}}]$ are both strongly consistent.*

Moreover, strong consistency is a sufficient condition for consistency (DeGroot and Schervish, 2012), hence all of TC-MINE, TC-NWJ and TC-InfoNCE estimators are also consistent statistics. For unbiasedness, though Theorem 3.3 indicates that unbiased MI estimators can lead to unbiased TC estimators, to the best of our knowledge, none of the previous variational MI estimators (Belghazi et al., 2018; Oord et al., 2018; Poole et al., 2019; Cheng et al., 2020a) are unbiased. Therefore, we leave the study of unbiased TC estimators for future work. Besides the above theoretic analysis, we empirically test our TC estimator in Section 5 with the application tasks introduced in Section 4.

4 RELATED WORK

Disentangled Representation Learning Disentangled representation learning (DRL) seeks to map each data instance into independent latent subspaces, while different subspaces meaningfully represent different attributes of the instance (Locatello et al., 2019b). Recently, DRL methods have attracted considerable interest on various learning tasks such as domain adaptation (Gholami et al., 2020), conditional generation (Burgess et al., 2018), and few-shot learning (Yuan et al., 2021). DRL methods are mainly recognized into two categories as unsupervised and supervised DRL. Prior unsupervised DRL works (Burgess et al., 2018; Kim and Mnih, 2018) utilize different regularizers to make each dimension of latent space to be as independent as possible, which has been challenged by Locatello et al. (2019b) in that each embedding dimension may not be related to a meaningful data attribute. Alternatively, supervised DRL methods (Hjelm et al., 2018; Kim and Mnih, 2018; Cheng et al., 2020b; Yuan et al., 2021) add different supervision terms on different embedding components, which effectively learn meaningful embedding while enabling disentanglement. Both supervised and unsupervised DRL methods require correlation reduction technique to prevent the embedding information from leaking into other embedding components. To reduce embedding correlation, Hjelm et al. (2018); Kim and Mnih (2018) use adversarial training methods, while Chen et al. (2018); Cheng et al. (2020b); Yuan et al. (2021) minimize statistical dependency (*i.e.*, MI and

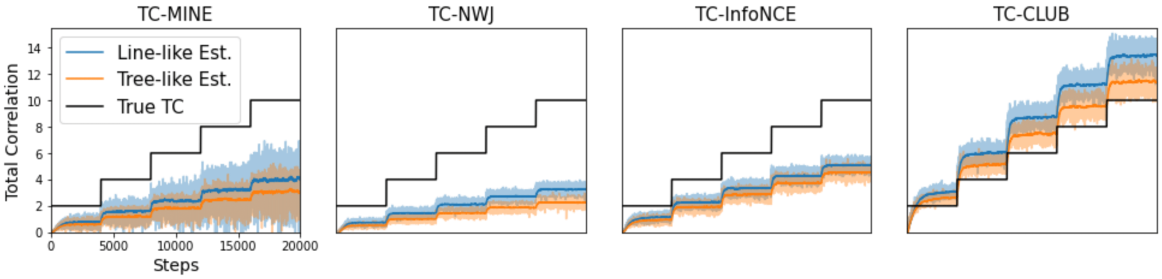


Figure 2: Simulation performance of TC *line-like* and *tree-like* estimators with different MI estimators.

TC), between different embedding components.

Contrastive Representation Learning Contrastive representation learning is a fundamental training methodology which maximizes the difference of positive and negative data pairs to obtain informative representations. In contrastive learning, a pairwise distance/similarity score function is always set to measure data pairs. Then, the learning objective is to maximize the margin between scores of positive and negative data pairs. Prior contrastive learning has achieved satisfying performance in numerous tasks, such as metric learning (Weinberger et al., 2006; Davis et al., 2007), word embedding learning (Mikolov et al., 2013), and graph embedding learning (Tang et al., 2015; Grover and Leskovec, 2016). Recently, contrastive learning has been recognized as a powerful tool in unsupervised or semi-supervised learning scenarios (He et al., 2020; Chen et al., 2020; Gao et al., 2019), which significantly narrows the gap of performance of supervised and unsupervised learning methods. Among these unsupervised methods, Gao et al. (2021) proposed a contrastive text representation learning framework (SimCSE). For each sentence, SimCSE use the dropout mechanism to generate sentence augmentation pairs, then maximize the mutual information within the augmented embedding pairs. The empirical results demonstrate contrastive learning obtains high-quality embeddings even with little supervision (Gao et al., 2021).

5 EXPERIMENTS

In this section, we empirically evaluate the effectiveness of our TCeld on three tasks: TC estimation, minimization, and maximization. For TC estimation, we generate synthetic data from correlated multi-variate Gaussian distributions, then compare the predictions from our TC estimator with the ground-truth TC values. For TC minimization, we conduct a multi-component disentangled representation learning experiment on Colored-MNIST (Esser et al., 2020) dataset, to minimize the total correlation among the digit, style, and color embeddings of digit images. To test the TC maximization ability, we apply our TC estimator into a contrastive text learning framework to maximize the TC value among different sentence augmentations. Since our proposed TC estimators can be flexibly induced by different MI estimators, for convenience, we refer the TC estimator as TC-(MI estimator name), or TC-(Line/Tree)-(MI estimator name) if the decomposition strategy is specified. For example, TC-Line-MINE denotes the TC estimator by line-like decomposition with the MINE MI estimator.

5.1 TC Estimation on Simulation Data

We first test the estimation quality of our TC estimators under simulation scenarios. We selected four MI estimators, MINE (Belghazi et al., 2018), NWJ (Nguyen et al., 2010), InfoNCE (Oord et al., 2018), and CLUB (Cheng et al., 2020a), to induce our TC estimators. Then we test TCeld with both tree-like and line-like strategies. The detailed description and implementation of the four MI estimators are shown in the Supplementary Material, where the results of non-variational methods like KDE, k-NN and their variants are also reported.

To evaluate TCeld's estimation performance with different ground-truth values, we sample simulated data from a four-dimensional Gaussian distributions $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a covariance matrix with all diagonal elements equal to 1, which means the variance of each $\boldsymbol{x}_i$ is normalized to 1. With this Gaussian assumption, the true TC value can be calculated in a closed-form as $\mathcal{TC}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4) = -\frac{1}{2} \log \operatorname{Det}(\boldsymbol{\Sigma})$, where $\operatorname{Det}(\boldsymbol{\Sigma})$ is the determinant of $\boldsymbol{\Sigma}$. Therefore, we can adjust the correlation coefficients (non-diagonal elements) in $\boldsymbol{\Sigma}$ to set the ground-truth TC values in the set $\{2.0, 4.0, 6.0, 8.0, 10.0\}$ (described in details in the Supplementary Material). The sample dimension is set to 20. The dimension of hidden states for variational estimators is 15. For each fixed TC value, we sample data 4000 times, with batch size 64 and learning rate 0.001 to train the estimators.

In Figures 2 we report the performance of TCeld with different MI bounds at each training step. In each figure, the true TC value is shown as a step function drawn as a black line. The line-like and tree-like estimation values are presented for different steps as shaded blue and orange curves respectively. The dark blue and orange curves illustrate the local averages of the estimated values, with a bandwidth equal to 200. With both tree-like and line-like decomposition, the estimation values for TC-MINE, TC-NWJ and TC-InfoNCE stay below the truth TC step functions, supporting the claims in Theorem 3.2. For the only MI upper bound CLUB, the

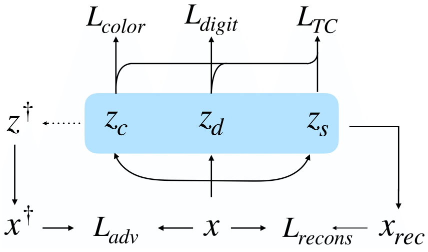


Figure 3: TC disentangling framework: $\boldsymbol{x}$ is an input image. $\boldsymbol{z}_c$, $\boldsymbol{z}_d$, $\boldsymbol{z}_s$ are the *color*, *digit*, *style* embeddings respectively. $\boldsymbol{x}_{\text{rec}}$ is the reconstruction. $\boldsymbol{x}^\dagger$ is a generated sample from the shuffled latent space for adversarial training.

estimated values initially lie beneath the ground-truth TC, but finally converge above the step function. This is because at the beginning of the estimator training, the parameters are not well learned from the synthetic samples, and fail to support a valid MI upper bound. With the training progress going on, the estimator performs better and finally converges to the desired upper-bound estimation.

Furthermore, we provided the bias, variance, and the mean squared error (MSE) of TC estimation values in the Supplementary Material. The tree-like and line-like strategies have insignificant effects on variance of our TC estimators, where TC-NWJ always keep the lowest variance. However, as for bias and MSE, TC-CLUB works uniformly better than other estimators under line-like step, while TC-InfoNCE outperforms others mostly under tree-like decomposition. By further analyzing this phenomenon, we find that when estimating $\mathcal{I}(\boldsymbol{v};\boldsymbol{u})$, CLUB requires a variational approximation $q_\theta(\boldsymbol{v}|\boldsymbol{u})$. When we use the line-like decomposition strategy, $\boldsymbol{v}=\boldsymbol{x}_{i+1}$ is always a single variable, and $\boldsymbol{u}=\boldsymbol{X}_{1:i}$ is the concatenation of $(\boldsymbol{x}_1,\dots,\boldsymbol{x}_i)$. The $q_\theta(\boldsymbol{v}|\boldsymbol{u})$ with a neural network implementation can have better performance with output $\boldsymbol{v}$ in a fixed low dimension. However, all the other MI estimators need to learn a score function $\phi(\boldsymbol{v},\boldsymbol{u})$, where the imbalanced inputs $\boldsymbol{v}=\boldsymbol{x}_{i+1}$ and $\boldsymbol{u}=\boldsymbol{X}_{1:i}$ can hinder the learning of function ϕ. In contrast, the tree-like TC estimators split variables equally into subgroups, which facilitate the learning of $\phi(\boldsymbol{u},\boldsymbol{v})$ with $\boldsymbol{u}=\boldsymbol{X}_{i:\lfloor(i+j)/2\rfloor}$ and $\boldsymbol{u}=\boldsymbol{X}_{\lfloor(i+j)/2\rfloor+1:j}$ for the lower-bound methods. For CLUB, the tree-like decomposition increases the output dimension of the variational net $q_\theta(\boldsymbol{v}|\boldsymbol{u})$ and makes the learning more challenging, explaining TC-CLUB's lower performance than TC-InfoNCE in Supplementary Material.

5.2 TC Minimization for Disentangled Representation Learning

For the TC minimization task, we conduct an experiment on a ColorMNIST hand-writing dataset (LeCun et al., 1998; Esser et al., 2020) to learn disentangled digit, color, and style representations of each hand-written number image. Each input data $(\boldsymbol{x}^i,\boldsymbol{y}_d^i,\boldsymbol{y}_c^i)$ contains three components: a image $\boldsymbol{x}^i$, its digit label $\boldsymbol{y}_d^i\in\{0,1,\dots,9\}$, and its color label vector $\boldsymbol{y}_c^i=(R^i,G^i,B^i)$, where $R^i,G^i,B^i\in[0,1]$ represents the intensity of colors (*Red*, *Green*, *Blue*).

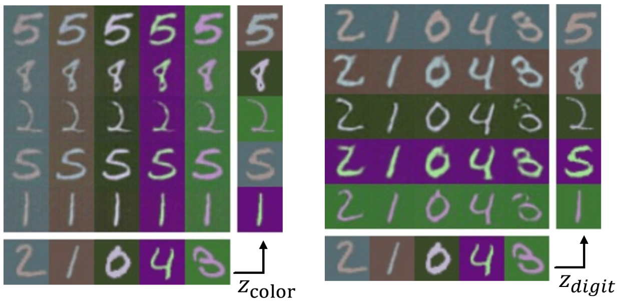


Figure 4: Generated examples with latent embedding swapping: The latent embeddings z_{color}(left-side), z_{digit}(right-side) of the bottom row (source) are swapped to the corresponding embeddings of images in the right column (target).

To learn the digit, color, and style representations from input images, we use the neural encoder $\mathcal{E}(\cdot)$ to map each image $\boldsymbol{x}_i$ to the corresponding latent representations $\boldsymbol{z}_d$, $\boldsymbol{z}_c$, $\boldsymbol{z}_s$, respectively. The digit representation $\boldsymbol{z}_d^i$ is supposed to include sufficient digit information from $\boldsymbol{x}^i$, hence we use a digit classifier $\mathcal{F}_d(\cdot)$ to predict the digit label with loss $\mathcal{L}_{digit}=\text{Cross-Entropy}(\mathcal{F}_d(\boldsymbol{z}_d^i),\boldsymbol{y}_d^i)$. Similarly, we set a color regression function $\mathcal{F}_c(\cdot)$ on the color embedding $\boldsymbol{z}_c^i$ to ensure representativeness by minimizing l-2 norm $\mathcal{L}_{\text{color}}=\|\mathcal{F}_c(\boldsymbol{z}_c^i)-\boldsymbol{y}_c^i\|^2$. Excluding the digit and color information, the remaining part from the image should be the human hand-writing style information, which is assumed to be independent of the digit and color information. Therefore, we minimize the total correlation $\mathcal{L}_{\text{TC}}=\mathcal{TC}(\boldsymbol{z}_d,\boldsymbol{z}_c,\boldsymbol{z}_s)$ as a regularizer to make sure different representation components do not include information from the others. Finally, we introduce a decoder $\mathcal{D}(\cdot)$ to reconstruct the original image $\boldsymbol{x}^i$ from $(\boldsymbol{z}_d,\boldsymbol{z}_c,\boldsymbol{z}_s)$ to induce sufficient information into the latent representations with loss $\mathcal{L}_{\text{recons}}=\|\mathcal{D}(\boldsymbol{z}_d,\boldsymbol{z}_c,\boldsymbol{z}_s)-\boldsymbol{x}\|^2$. To further enhance the generation quality of the decoder $\mathcal{D}(\cdot)$, we adapt an adversarial learning regularizer, where we randomize the combination of the latent representation components in each batch, then treat the corresponding decoder output as artificial (synthesized) data $\boldsymbol{x}^\dagger$. With a discriminator $\mathcal{H}$, we use adversarial training (Goodfellow et al., 2014) to ensure the decoder to generate high-quality samples with loss $\mathcal{L}_{\text{adv}}=\mathbb{E}_{\boldsymbol{x}}[\log\mathcal{H}(\boldsymbol{x})]+\mathbb{E}_{\boldsymbol{x}^\dagger}[\log(1-\mathcal{H}(\boldsymbol{x}^\dagger))]$. Figure 3 illustrates the whole framework. More details about the framework are shown in the Supplementary Material.

Evaluations To evaluate the quality of our disentangled representations, we conduct a controllable generation testing task to check whether the learned embedding components ($\boldsymbol{z}_d$, $\boldsymbol{z}_c$, or $\boldsymbol{z}_s$) can control the corresponding attributes (digit, color, or style) of the generated sample $\mathcal{D}(\boldsymbol{z}_d,\boldsymbol{z}_c,\boldsymbol{z}_s)$. Hence, we consider three perspectives to evaluate the disentanglement: ($\boldsymbol{i}$) *digit transfer*: Select another real sample

Table 1: Controllable generation results on ColorMNIST. "✓" means adversary training is applied. Acc$_d$ and Acc$_c$ are generated digit classification accuracy. Res.l_2 measures the residual l_2-distance for color transferred samples.

Adv.	Methods	Acc$_d$	Acc$_c$	Res. l_2
	AE	2.70	91.65	168.95
	TC (Ours)	92.70	97.69	**28.66**
✓	AE	0.14	**98.82**	389.73
✓	VAE	20.24	92.21	254.83
✓	DIIN	94.57	97.45	79.38
✓	TC (Ours)	**96.38**	98.23	43.04

$\boldsymbol{x}'$ from testing set and obtain its digit embedding $\boldsymbol{z}'_d$. Next, feed the new latent embedding combination $(\boldsymbol{z}'_d, \boldsymbol{z}_c, \boldsymbol{z}_s)$ into the decoder to generate a sample $\boldsymbol{x}^\dagger = \mathcal{D}(\boldsymbol{z}'_d, \boldsymbol{z}_c, \boldsymbol{z}_s)$, which is supposed to share the same digit information with $\boldsymbol{x}'$, so we predict the digit label on $\boldsymbol{x}^\dagger$ then report the classification accuracy Acc$_d$. $(\boldsymbol{ii})$ *digit preservation*: Select another testing sample $\boldsymbol{x}''$, and replace $\boldsymbol{z}_c$ with $\boldsymbol{z}''_c$ to generate $\boldsymbol{x}^\ddagger = \mathcal{D}(\boldsymbol{z}_d, \boldsymbol{z}''_c, \boldsymbol{z}_s)$. Then predict the digit label of $\boldsymbol{x}^\ddagger$ and report the digit classification accuracy Acc$_c$. $(\boldsymbol{iii})$ *color transfer*: For selected $\boldsymbol{x}''$ in (ii), we have its color label vector $\boldsymbol{y}''_c$. Hence, we can directly synthesize a $\bar{\boldsymbol{x}}^\ddagger$ by setting $\boldsymbol{y}''_c$ on $\boldsymbol{x}$. We report the l_2-distance (Res.l_2) between generated $\boldsymbol{x}^\ddagger$ and synthetic $\bar{\boldsymbol{x}}^\ddagger$ as color transfer quality.

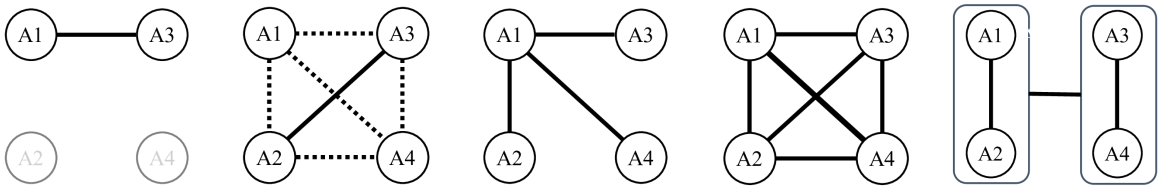
Results and Analysis We compared our method with vanilla auto-encoder (AE), variational auto-encoder (VAE) (Kingma and Welling, 2013), and DIIN (Esser et al., 2020), and report the aforementioned evaluation metrics in Table 1. Implementation and setup details are provided in the Supplementary Material. Vanilla AE fails on Acc$_d$ because without any disentangled regularizer, $\boldsymbol{z}_s$ can contain the information revealed from $\boldsymbol{z}_d$ and $\boldsymbol{z}_c$ VAE partially solves the embedding entanglement problem with its KL divergence term in the learning objective, which encourages the latent embedding being close to a dimension-wise-independent standard Gaussian. DIIN(Esser et al., 2020) achievers more significant improvement, by projecting the latent space of an auto-encoder to a multi-variate Gaussian distribution using normalizing flow (Kingma and Dhariwal, 2018). Our TC-based method uniformly outperforms the vanilla AE without the adversarial training. Among all method with adversarial training, the vanilla AE reaches the highest color classification accuracy, for which our TC method is also strongly competitive. Moreover, our TC is in the lead on the other two metrics and leave a significant performance gap to the vanilla AE, which indicates our TC-based method can learn more balanced attribute embeddings in terms of representativeness and disentanglement. In addition, we show the generated image examples of digit transfer (i) and color transfer (iii) in Figure 4, where both color and digit information can be successfully preserved in the transferred images.

5.3 TC Maximization for Contrastive Representation Learning

To test the performance of TCeld on TC maximization, we conduct a unsupervised text representation learning experiment following the SimCSE (Gao et al., 2021) setups. More specifically, we aim to train a encoder $\mathcal{E}(\cdot)$ to map each sentence $\boldsymbol{x}$ into representative embedding $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x})$. According to SimCSE (introduced in Section 4), one can learn in a unsupervised way the encoder $\mathcal{E}(\cdot)$, by first generating several data augmentations $(\tilde{\boldsymbol{x}}_1, \tilde{\boldsymbol{x}}_2, \dots, \tilde{\boldsymbol{x}}_n)$, then maximizing the correlation of corresponding $(\tilde{\boldsymbol{z}}_1, \tilde{\boldsymbol{z}}_2, \dots, \tilde{\boldsymbol{z}}_n)$. Most of the previous contrastive learning methods (Chen et al., 2020; Gao et al., 2021; Jiang et al., 2022) focus on two-augmentation cases, where the mutual information between the two augmentations $\mathcal{I}(\tilde{\boldsymbol{z}}_1; \tilde{\boldsymbol{z}}_2)$ is maximized for each input $\boldsymbol{x}$. Moreover, none of current text contrastive learning method handles multi-augmentation ($n \geq 3$) situations. However, Tian et al. (2020) point out that contrastive learning with more augmentations can further enhance the latent embedding quality. Therefore, for testing our TC estimators while attempting the *first* multi-augmentation text contrastive learning, we plan to maximize $\mathcal{TC}(\tilde{\boldsymbol{z}}_1, \tilde{\boldsymbol{z}}_2, \dots, \tilde{\boldsymbol{z}}_n)$ to train the text encoder $\mathcal{E}$.

Model Frameworks We conduct our multi-augmentation text contrastive leaning by extending prior two-augmentation methods, SimCSE (Gao et al., 2021) and PromptBERT (Jiang et al., 2022) with *four* augmentations for each text input. Both SimCSE and PromptBERT maximize the InfoNCE MI estimator between the two augmentation embeddings with a BERT(Devlin et al., 2018)-based pretrained text encoder. Correspondingly, our extended SimCSE-TC and PromptBERT-TC utilize the same encoder structure but maximize TC-InfoNCE of four augmentation embeddings for each sentence. Following SimCSE (Gao et al., 2021) and PromptBERT (Jiang et al., 2022), we finetine the text encoder $\mathcal{E}$ on pretrained BERT$_{\text{base}}$ (Devlin et al., 2018) and RoBERTa$_{\text{base}}$ (Liu et al., 2019). Based on our observation in Section 5.1, tree-like decomposition empirically works better for InfoNCE estimator. Therefore, we select TC-Tree-InfoNCE for this text contrastive learning task. More setup details can be found in the Supplementary Material.

Evaluation Following previous work (Gao et al., 2021; Jiang et al., 2022), we evaluate models on 7 semantic textual similarity (STS) datasets: STS12 (Agirre et al., 2012), STS13 (Agirre et al., 2013), STS14 (Agirre et al., 2014), STS15 (Agirre et al., 2015), STS16 (Agirre et al., 2016), STS Benchmark (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). The task is to predict the similarity (ranging from 0 to 5) between paired sentences with learned text representations. We report Spearman's correlation (Myers et al., 2013) between model prediction and the ground truth. More details about the model design, hyperparameter settings and evaluation metrics are in the



(a) MI-fix (b) MI-sample (c) Core-View (d) Full-Graph (e) TC-Tree

Figure 5: Correlation maximization strategy for multiple augmentations. Nodes A_1, A_2, A_3, A_4 mean four input augmentations. Each solid line indicates an MI maximization between the connected two augmentations. Each dashed line means the connected augmentation pair is randomly selected.

Table 2: Text representation evaluation on STS tasks (Spearman's correlation with "all" setting). Methods with " -TC" means that total correlation is used as the loss function.

Method	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
IS-BERT$_{\text{base}}$ (Zhang et al., 2020)	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
ConSERT$_{\text{base}}$ (Yan et al., 2021)	64.64	78.49	69.07	79.72	75.95	73.97	67.31	72.74
SimCSE-BERT$_{\text{base}}$ (Gao et al., 2021)	68.40	82.41	74.38	80.91	78.56	76.85	72.23	76.25
SimCSE-BERT$_{\text{base}}$-TC	68.66	81.45	74.34	81.81	79.24	78.85	72.46	76.69
PromptBERT$_{\text{base}}$ (Jiang et al., 2022)	71.56	84.58	76.98	84.47	**80.60**	81.60	69.87	$78.54_{\pm 0.15}$
PromptBERT$_{\text{base}}$-TC	**72.05**	**84.61**	**77.23**	**84.73**	80.34	**81.89**	**70.23**	**78.72**$_{\pm 0.10}$
SimCSE-RoBERTa$_{\text{base}}$ (Gao et al., 2021)	70.16	81.77	73.24	81.36	80.65	80.22	68.56	76.57
SimCSE-RoBERTa$_{\text{base}}$-TC	71.46	82.16	74.14	82.17	80.93	80.02	68.24	77.02
PromptRoBERTa$_{\text{base}}$ (Jiang et al., 2022)	**73.94**	84.74	77.28	84.99	81.74	81.88	69.50	$79.15_{\pm 0.25}$
PromptRoBERTa$_{\text{base}}$-TC	72.58	**85.06**	**78.24**	**85.82**	**81.95**	**82.94**	**70.47**	**79.58**$_{\pm 0.25}$

Table 3: Ablation study of different correlation maximization strategies. Average Spearman coefficient is reported over the seven STS tasks.

Model	Prompt	Prompt
Method	BERT	RoBERTa
MI-fix	$78.54_{\pm 0.15}$	$79.28_{\pm 0.23}$
MI-sample	$78.41_{\pm 0.13}$	$79.27_{\pm 0.34}$
Core-View (Tian et al., 2020)	$78.64_{\pm 0.22}$	$79.41_{\pm 0.25}$
Full-Graph (Tian et al., 2020)	$78.52_{\pm 0.13}$	$79.49_{\pm 0.26}$
TC	$78.72_{\pm 0.10}$	$79.58_{\pm 0.25}$

Supplementary Material.

Results and Analysis We report the mean and standard deviation over 10 runs with different random seeds in Table 2. On most of the evaluation datasets, our TC-based methods outperform their corresponding baselines, in which PromptBERT/RoBERTa are the state-of-the-art unsupervised sentence representation learning methods. These results also underline the claim that more augmentations lead to higher representation quality in contrastive learning.

For the ablation study, we fix the number of augmentations to 4, and test the influence of different embedding correlation maximization strategies. Since there is no prior work on multi-augmentation text contrastive learning, we proposed several substitute strategies by ourselves in Figure 5: (a) *MI-fix*: only select the first two augmentations and omit the others; (b) *MI-sample*: randomly select two augmentations and omit the others; (c) *Core-View* (Tian et al., 2020): calculate the MI between one fixed augmentation and each augmentation in the rest. (d) *Full-Graph* (Tian et al., 2020): calculate the MI values between each augmentation pairs. From the results in Table 3, MI-fix and MI-sample which only utilized two augmentations, have lower Spearman score than the other methods using all augmentations' information. Our TC-based correlation maximization strategy slightly outperforms Core-View and Full-Graph. Core-View does not consider the correlation among the other augmentations (A_2, A_3, A_4 in Figure 5c). Full-Graph uses expensive MI estimators, which quadratically increases the computational complexity of augmentation correlation maximization, while being more prone to overfitting than TC with the same data size.

6 CONCUCLUSION

We have derived a Total Correlation Estimation with Linear Decomposition (TCeld), which converts total correlation into summation of mutual information terms along two separation paths (*i.e.*, line-like and tree-like). By applying variational estimators to each MI term in TCeld, we have obtained TC-Line and TC-Tree estimators. Further, we have analyzed the statistical properties of the proposed TC estimators and claimed their strong consistency when induced by appropriate MI estimators such as MINE, NWJ, and InfoNCE. Moreover, we have empirically demonstrated the effectiveness of the proposed TC estimators on both TC estimation and optimization tasks. The experimental results show that our TC estimators can only provide reliable estimation from samples, but also serve as an effective learning regularizer for model training. More properties of TCeld, such as sample complexity, unbiasedness, and low-variance, remain to be explored both theoretically and empirically. We hope this study can serve to promote TC, as a multi-variate information concept, to apply into cutting-edge deep learning models, such as representation learning, controllable generation, model distillation and ensemble.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, 2012.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43, 2013.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91, 2014.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 252–263, 2015.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics), 2016.

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

David Barber and Felix V Agakov. The im algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, page None, 2003.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Devon Hjelm, and Aaron Courville. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 530–539, 2018.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Jean-François Cardoso. Dependence, correlation and gaussianity in independent component analysis. *Journal of Machine Learning Research*, 4(Dec):1177–1203, 2003.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4791–4798. IEEE, 2015.

Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

Pengyu Cheng and Ruineng Li. Replacing language model for style transfer. *arXiv preprint arXiv:2211.07343*, 2022.

Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. CLUB: A contrastive log-ratio upper bound of mutual information. *arXiv preprint arXiv:2006.12013*, 2020a.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. Improving disentangled text representation learning with information-theoretic guidance. *arXiv preprint arXiv:2006.00693*, 2020b.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. Fairfil: Contrastive neural debiasing method for pretrained text encoders. In *International Conference on Learning Representations*, 2021.

Elliot Creager, David Madras, Joern-Henrik Jacobsen, Marissa Weis, Kevin Swersky, Toniann Pitassi, and Richard Zemel. Flexibly fair representation learning by disentanglement. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1436–1445. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/creager19a.html`.

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007.

Morris H DeGroot and Mark J Schervish. *Probability and statistics*. Pearson Education, 2012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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.

Patrick Esser, Robin Rombach, and Bjorn Ommer. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232, 2020.

Wikimedia Foundation. Wikimedia downloads. URL `https://dumps.wikimedia.org`.

Shuyang Gao, Rob Brekelmans, Greg Ver Steeg, and Aram Galstyan. Auto-encoding total correlation explanation. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1157–1166, 2019.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, 2021.

Weihao Gao, Sewoong Oh, and Pramod Viswanath. Demystifying fixed k-nearest neighbor information estimators. *IEEE Transactions on Information Theory*, 64(8):5629–5661, 2018.

Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Clive Granger and Jin-Lung Lin. Using the mutual information coefficient to identify lags in nonlinear models. *Journal of time series analysis*, 15(4):371–384, 1994.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

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, 2010.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

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.

Bo Jiang, Chao Ye, and Jun S Liu. Nonparametric k-sample tests via dynamic slicing. *Journal of the American Statistical Association*, 110(510):642–653, 2015.

Ting Jiang, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Liangjie Zhang, and Qi Zhang. Promptbert: Improving BERT sentence embeddings with prompts. *CoRR*, abs/2201.04337, 2022. URL `https://arxiv.org/abs/2201.04337`.

Brian J Julian, Sertac Karaman, and Daniela Rus. On mutual information-based control of range sensing robots for mapping applications. *The International Journal of Robotics Research*, 33(10):1375–1392, 2014.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Barnabas Poczos, Larry Wasserman, et al. Nonparametric von mises estimators for entropies, divergences and mutual informations. *Advances in Neural Information Processing Systems*, 28, 2015.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 69 6 Pt 2:066138, 2004a.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004b.

Alexander Lachmann, Federico M Giorgi, Gonzalo Lopez, and Andrea Califano. Aracne-ap: gene network reverse engineering through adaptive partitioning inference of mutual information. *Bioinformatics*, 32(14):2233–2235, 2016.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. In *Advances in Neural Information Processing Systems*, pages 14611–14624, 2019a.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124, 2019b.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. A sick cure for the evaluation of compositional distributional semantic models. In *Lrec*, pages 216–223. Reykjavik, 2014.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Jerome L Myers, Arnold D Well, and Robert F Lorch Jr. *Research design and statistical analysis*. Routledge, 2013.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Dávid Pál, Barnabás Póczos, and Csaba Szepesvári. Estimation of rényi entropy and mutual information based on generalized nearest-neighbor graphs. *Advances in Neural Information Processing Systems*, 23, 2010.

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

Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5102–5112. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/peng19b.html`.

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, 2019.

Shashank Singh and Barnabás Póczos. Exponential concentration of a density functional estimator. *Advances in Neural Information Processing Systems*, 27, 2014.

Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2164–2173, 2019.

Milan Studenỳ and Jirina Vejnarová. The multi-information function as a tool for measuring stochastic dependence. In *Learning in graphical models*, pages 261–297. Springer, 1998.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020.

Greg Ver Steeg and Aram Galstyan. Discovering structure in high-dimensional data through correlation explanation. *Advances in Neural Information Processing Systems*, 27, 2014.

Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of research and development*, 4(1):66–82, 1960.

Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*, 2021.

Siyang Yuan, Pengyu Cheng, Ruiyi Zhang, Weituo Hao, Zhe Gan, and Lawrence Carin. Improving zero-shot voice style transfer via disentangled representation learning. In *International Conference on Learning Representations*, 2021.

Diego J Zea, Diego Anfossi, Morten Nielsen, and Cristina Marino-Buslje. Mitos. jl: mutual information tools for protein sequence analysis in the julia language. *Bioinformatics*, 33(4):564–565, 2016.

Yan Zhang, Ruidan He, Zuozhu Liu, Kwan Hui Lim, and Lidong Bing. An unsupervised sentence embedding method by mutual information maximization. *arXiv preprint arXiv:2009.12061*, 2020.

Sicheng Zhu, Xiao Zhang, and David Evans. Learning adversarially robust representations via worst-case mutual information maximization. *arXiv preprint arXiv:2002.11798*, 2020.

A PROOFS

Proof of Theorem 3.1. Note that $\boldsymbol{X}_{\mathcal{A}} := (\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \dots, \boldsymbol{x}_{i_m})$ and $\boldsymbol{X}_{\hat{\mathcal{A}}} = \boldsymbol{X}/\boldsymbol{X}_{\mathcal{A}}$. Denote $\boldsymbol{X}_{\hat{\mathcal{A}}} = (\boldsymbol{x}_{j_1}, \boldsymbol{x}_{j_2}, \dots, \boldsymbol{x}_{j_l})$. Then

$$\begin{aligned}
&\mathcal{TC}(\boldsymbol{X}) = \mathbb{E}_{p(\boldsymbol{X})}\left[\log \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)}{p(\boldsymbol{x}_1)p(\boldsymbol{x}_2)\dots p(\boldsymbol{x}_n)}\right] \\
=&\mathbb{E}_{p(\boldsymbol{X})}\left[\log\left(\frac{p(\boldsymbol{X}_{\mathcal{A}})}{p(\boldsymbol{x}_{i_1})p(\boldsymbol{x}_{i_2})\dots p(\boldsymbol{x}_{i_m})} \cdot \frac{p(\boldsymbol{X}_{\hat{\mathcal{A}}})}{p(\boldsymbol{x}_{j_1})p(\boldsymbol{x}_{j_2})\dots p(\boldsymbol{x}_{j_l})} \cdot \frac{p(\boldsymbol{X})}{p(\boldsymbol{X}_{\mathcal{A}})p(\boldsymbol{X}_{\hat{\mathcal{A}}})}\right)\right] \\
=&\mathcal{TC}(\boldsymbol{X}_{\mathcal{A}}) + \mathcal{TC}(\boldsymbol{X}_{\hat{\mathcal{A}}}) + \mathcal{I}(\boldsymbol{X}_{\mathcal{A}}; \boldsymbol{X}_{\hat{\mathcal{A}}})
\end{aligned}$$

□

Proof of Corollary 3.1.2. We denote $\boldsymbol{X}_{i:j} := (\boldsymbol{x}_i, \boldsymbol{x}_{i+1}, \dots, \boldsymbol{x}_{j-1}, \boldsymbol{x}_j)$. Note that

$$\begin{aligned}
\mathcal{TC}(\boldsymbol{X}_{1:n}) =&\mathbb{E}_{p(\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_n)}\left[\log \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n)}{p(\boldsymbol{x}_1)p(\boldsymbol{x}_2)\dots p(\boldsymbol{x}_n)}\right] \\
=&\mathbb{E}_{p(\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_n)}\left[\log\left(\frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{n-1}, \boldsymbol{x}_n)}{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{n-1})p(\boldsymbol{x}_n)} \cdot \frac{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{n-1})}{p(\boldsymbol{x}_1)p(\boldsymbol{x}_2)\dots p(\boldsymbol{x}_{n-1})}\right)\right] \\
=&\mathcal{I}(\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{n-1}; \boldsymbol{x}_n) + TC(\boldsymbol{X}_{1:n-1}) \\
=&\mathcal{I}(\boldsymbol{X}_{1:n-1}; \boldsymbol{x}_n) + \mathcal{TC}(\boldsymbol{X}_{1:n-1})
\end{aligned}$$

Similarly,

$$\mathcal{TC}(\boldsymbol{X}_{1:n}) = \mathcal{I}(\boldsymbol{X}_{1:n-1}; \boldsymbol{x}_n) + \mathcal{I}(\boldsymbol{X}_{1:n-2}; \boldsymbol{x}_{n-1}) + \mathcal{TC}(\boldsymbol{X}_{1:n-2}) = \sum_{i=1}^{n-1} \mathcal{I}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1}) \tag{11}$$

□

Proof of Theorem 3.3. First consider line-like TC estimator $\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}](\boldsymbol{X}) = \sum_{i=1}^{n-1} \hat{\mathcal{I}}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1})$. If the MI estimator $\hat{\mathcal{I}}$ is unbiased, by definition, $\mathbb{E}[\hat{\mathcal{I}}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1})] = \mathcal{I}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1})$. Taking expectation for the TC estimator,

$$\mathbb{E}[\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}](\boldsymbol{X})] = \sum_{i=1}^{n-1} \mathbb{E}[\hat{\mathcal{I}}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1})] = \sum_{i=1}^{n-1} \mathcal{I}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1}) = \mathcal{TC}(\boldsymbol{X}), \tag{12}$$

which means that $\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}](\boldsymbol{X})$ is unbiased. Similarly, we can show that $\widehat{\mathcal{TC}}_{\text{Tree}}[\hat{\mathcal{I}}](\boldsymbol{X})$ is unbiased. □

Proof of Theorem 3.4. For the convenience of notation, we show the proof with our $\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]$ estimator. The proof can be easily applied to $\hat{\mathcal{TC}}_{\text{Tree}}[\hat{\mathcal{I}}]$, since both $\hat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]$ and $\hat{\mathcal{TC}}_{\text{Tree}}[\hat{\mathcal{I}}]$ are linear combination of MI terms. We denote $\hat{\mathcal{I}}_m(\boldsymbol{x}; \boldsymbol{y}) = \hat{\mathcal{I}}(\{(\boldsymbol{x}^k, \boldsymbol{y}^k)\}_{k=1}^m)$ and $\widehat{\mathcal{TC}}[\hat{\mathcal{I}}]_m(\boldsymbol{X}) = \widehat{\mathcal{TC}}[\hat{\mathcal{I}}](\{\boldsymbol{X}^k\}_{k=1}^m)$ as the estimators with m samples $\{(\boldsymbol{x}^k, \boldsymbol{y}^k)\}_{k=1}^m \sim p(\boldsymbol{x}, \boldsymbol{y})$, and $\{\boldsymbol{X}^k\}_{k=1}^m \sim p(\boldsymbol{X})$ respectively.

Strong Consistency: If $\hat{\mathcal{I}}$ is a strong consistent estimator, by the Definition 2.3, $\forall \varepsilon > 0$, with a fix variable dimension $n \in \mathbb{N}_+$, for each variable pair $(\boldsymbol{X}_{1:i}, \boldsymbol{x}_{i+1})$ $(i = 1, 2, \dots, n-1)$, $\exists M_i > 0$, such that $\forall m > M_i$,

$$\mathbb{P}\Big\{ \Big|\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1}) - \mathcal{I}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1})\Big| \le \frac{\varepsilon}{n}\Big\} = 1. \tag{13}$$

Let $\bar{M} = \max\{M_1, M_2, \dots, M_{n-1}\}$, then $\forall m > \bar{M}$,

$$\mathbb{P}\Big\{ \Big|\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]_m(\boldsymbol{X}) - \mathcal{TC}(\boldsymbol{X})\Big| \le \varepsilon\Big\} \tag{14}$$

$$=\mathbb{P}\Big\{ \Big|\sum_{i=1}^{n-1} \hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1}) - \sum_{i=1}^{n-1} \mathcal{I}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1})\Big| \le \varepsilon\Big\} \tag{15}$$

$$\ge\mathbb{P}\left[\bigcap_{i=1}^{n-1} \Big\{ \Big|\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1}) - \mathcal{I}(\boldsymbol{X}_{1:i}; \boldsymbol{x}_{i+1})\Big| \le \frac{\varepsilon}{n}\Big\}\right]. \tag{16}$$

The inequality between equation 15 and equation 16 is because the condition in equation 15 is sufficient to deduce the condition in equation 15:

$$\begin{aligned}&\left|\sum_{i=1}^{n-1}\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})-\sum_{i=1}^{n-1}\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})\right|\\ \leq&\sum_{i=1}^{n-1}\left|\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})-\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})\right|\leq\frac{n-1}{n}\varepsilon<\varepsilon.\end{aligned}$$

Denote event $\mathcal{B}_i=\left\{\left|\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})-\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})\right|\leq\frac{\varepsilon}{n}\right\}$, by equation 13, $\mathbb{P}[\mathcal{B}_i]=1$. Consider the union $\mathcal{B}_i\cup\mathcal{B}_j$, we have:

$$1=\mathbb{P}[\mathcal{B}_i]\leq\mathbb{P}[\mathcal{B}_i\cup\mathcal{B}_j]=\mathbb{P}[\mathcal{B}_i]+\mathbb{P}[\mathcal{B}_j]-\mathbb{P}[\mathcal{B}_i\cap\mathcal{B}_j]=2-\mathbb{P}[\mathcal{B}_i\cap\mathcal{B}_j]\leq 1,\tag{17}$$

which means $\mathbb{P}[\mathcal{B}_i\cap\mathcal{B}_j]=1$. Iteratively applying this conclusion, we know $\mathbb{P}[\cap_{i=1}^{n-1}\mathcal{B}_i]=1$. Therefore, $\mathbb{P}\left[\bigcap_{i=1}^{n-1}\left\{\left|\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})-\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})\right|\leq\frac{\varepsilon}{n}\right\}\right]=\mathbb{P}[\cap_{i=1}^{n-1}\mathcal{B}_i]=1$. Combining with equation 14 and equation 16, we conclude that $\forall\varepsilon$, $\exists\hat{M}$, such that $\forall m>\hat{M}$, $\left|\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]_m(\boldsymbol{X})-\mathcal{TC}(\boldsymbol{X})\right|\leq\varepsilon$ almost surely, which supports $\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]$ is strongly consistent.

Consistency: If $\hat{\mathcal{I}}$ is a consistent estimator, by the Definition 2.2, $\forall\varepsilon>0$ and $\sigma>0$, with a fixed variable dimension $n\in\mathbb{N}_+$, for each variable pair $(\boldsymbol{X}_{1:i},\boldsymbol{x}_{i+1})$ $(i=1,2,\ldots,n-1)$, $\exists M_i$, such that $\forall m>M_i$,

$$\mathbb{P}\Big\{\left|\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})-\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})\right|<\frac{\varepsilon}{n}\Big\}>1-\frac{\sigma}{n}.\tag{18}$$

Let $\hat{M}=\max\{M_1,M_2,\ldots,M_{n-1}\}$, then $\forall m>\hat{M}$, similar to equation 14, equation 15, and equation 16,

$$\begin{aligned}&\mathbb{P}\Big\{\left|\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]_m(\boldsymbol{X})-\mathcal{TC}(\boldsymbol{X})\right|<\varepsilon\Big\}&(19)\\ =&\mathbb{P}\Big\{\left|\sum_{i=1}^{n-1}\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})-\sum_{i=1}^{n-1}\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})\right|<\varepsilon\Big\}&(20)\\ \geq&\mathbb{P}\left[\bigcap_{i=1}^{n-1}\Big\{\left|\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})-\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})\right|<\frac{\varepsilon}{n}\Big\}\right].&(21)\end{aligned}$$

Denote $\mathcal{B}_i=\left\{\left|\hat{\mathcal{I}}_m(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})-\mathcal{I}(\boldsymbol{X}_{1:i};\boldsymbol{x}_{i+1})\right|<\frac{\varepsilon}{n}\right\}$. Since $1\geq\mathbb{P}[\mathcal{B}_i\cup\mathcal{B}_j]=\mathbb{P}[\mathcal{B}_i]+\mathbb{P}[\mathcal{B}_j]-\mathbb{P}[\mathcal{B}_i\cap\mathcal{B}_j]$, we have $\mathbb{P}[\mathcal{B}_i\cap\mathcal{B}_j]\geq\mathbb{P}[\mathcal{B}_i]+\mathbb{P}[\mathcal{B}_j]-1>(1-\frac{\sigma}{n})+(1-\frac{\sigma}{n})-1=1-\frac{2}{n}\sigma$. Similarly, we can obtain $\mathbb{P}[\mathcal{B}_i\cap\mathcal{B}_j\cap\mathcal{B}_k]\geq\mathbb{P}[\mathcal{B}_i\cap\mathcal{B}_j]+\mathbb{P}[\mathcal{B}_k]-1\geq(1-\frac{2}{n}\sigma)+(1-\frac{\sigma}{n})=1-\frac{3}{n}\sigma$ and $\mathbb{P}\{\left|\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]_m(\boldsymbol{X})-\mathcal{TC}(\boldsymbol{X})\right|<\varepsilon\}\geq\mathbb{P}\left[\cap_{i=1}^{n-1}\mathcal{B}_i\right]\geq 1-\sigma$. Therefore, $\forall\varepsilon>0$, $\lim_{m\to\infty}\mathbb{P}\{|\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]_m(\boldsymbol{X})-\mathcal{TC}(\boldsymbol{X})|\geq\varepsilon\}=0$. $\widehat{\mathcal{TC}}_{\text{Line}}[\hat{\mathcal{I}}]$ is consistent. □

Proof of Corollary 3.4.2. By Theorem 3.4, only need to show both $\hat{\mathcal{I}}_{\text{InfoNCE}}$ and $\hat{\mathcal{I}}_{\text{NWJ}}$ are strongly consistent. Inspired by the proof of Theorem 2 in Belghazi et al. (2018), we only need to proof the following two lemmas:

Lemma A.1. *For any $\eta>0$, there exists a feedforward score network function $\hat{\phi}:\Omega\to\mathbb{R}$ such that $|\mathcal{I}(\boldsymbol{x},\boldsymbol{y})-\hat{\mathcal{I}}[\hat{\phi}]|\leq\eta$, where $\hat{\mathcal{I}}\in\{\hat{\mathcal{I}}_{\mathit{InfoNCE}},\hat{\mathcal{I}}_{\mathit{NWJ}}\}$, $\hat{\mathcal{I}}_{\mathit{InfoNCE}}[\hat{\phi}]=\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\hat{\phi}(\boldsymbol{x},\boldsymbol{y})]-\mathbb{E}_{p(\boldsymbol{x})}[\log\mathbb{E}_{p(\boldsymbol{y})}[\exp\hat{\phi}(\boldsymbol{x},\boldsymbol{y})]]$ and $\hat{\mathcal{I}}_{\mathit{NWJ}}[\hat{\phi}]=\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\hat{\phi}(\boldsymbol{x},\boldsymbol{y})]-\mathbb{E}_{p(\boldsymbol{x})p(\boldsymbol{y})}[\exp(\hat{\phi}(\boldsymbol{x},\boldsymbol{y})-1)]$*

Lemma A.2. *For any $\eta>0$, let $\mathcal{H}$ be the family of functions $\phi:\Omega\to\mathbb{R}$ defined by a give network architecture. Assume the parameter θ of network ϕ are restricted to some compact domain $\Theta\subset\mathbb{R}^k$. Then there exists $N\in\mathbb{N}_+$, such that, $\forall m\geq N$, $|\hat{\mathcal{I}}_m(\boldsymbol{x},\boldsymbol{y})-\sup_{\phi\in\mathcal{H}}\hat{\mathcal{I}}[\phi]|\leq\eta$ with probability one. Here $\hat{\mathcal{I}}\in\{\hat{\mathcal{I}}_{\mathit{InfoNCE}},\hat{\mathcal{I}}_{\mathit{NWJ}}\}$.*

To proof Lemma A.1, for NWJ, we select function $\phi^*(\boldsymbol{x},\boldsymbol{y})=1+\log\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}$. Then $\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\hat{\mathcal{I}}_{\text{InfoNCE}}[\phi^*]]=\mathcal{I}(\boldsymbol{x},\boldsymbol{y})$. The difference can be calculated as

$$\mathcal{I}(\boldsymbol{x},\boldsymbol{y})-\hat{\mathcal{I}}_{\text{NWJ}}[\phi]=\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\phi^*-\phi]-\exp(-1)\mathbb{E}_{p(\boldsymbol{x})p(\boldsymbol{y})}[\exp(\phi^*)-\exp(\phi)].\tag{22}$$

The right-hand side of equation 22 has the same form as equation (25) in Belghazi et al. (2018), except a coefficient $\exp(-1)$ for the second term. Therefore, we can exactly follow the proof of Section 6.2.1 of Belghazi et al. (2018) to prove our Lemma A.1 for the NWJ estimator, with only adjusting the coefficient weight of term $|\phi^* - \phi|$ and term $|\exp(\phi^*) - \exp(\phi)|$.

For InfoNCE, we select $\phi^*(\boldsymbol{x}, \boldsymbol{y}) = \log p(\boldsymbol{x}|\boldsymbol{y})$, so that $\mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\hat{\mathcal{I}}_{\text{InfoNCE}}[\phi^*]] = \mathcal{I}(\boldsymbol{x}; \boldsymbol{y})$. The difference can be written as

$$\mathcal{I}(\boldsymbol{x}; \boldsymbol{y}) - \hat{\mathcal{I}}_{\text{InfoNCE}}[\phi] = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})}[\phi^* - \phi] - \mathbb{E}_{p(\boldsymbol{x})}[\log \mathbb{E}_{p(\boldsymbol{y})}[\exp(\phi^*)] - \log \mathbb{E}_{p(\boldsymbol{y})}[\exp(\phi)]]. \tag{23}$$

Similarly to Section 6.2.1 in (Belghazi et al., 2018), we can consider the cases whether ϕ is bounded, then apply the universal approximation theorem to show Lemma A.1 for InfoNCE.

To proof Lemma A.2, we denote $\mathbb{P} = p(\boldsymbol{x}, \boldsymbol{y}), \mathbb{Q} = p(\boldsymbol{x})p(\boldsymbol{y})$, and $\mathbb{P}_m, \mathbb{Q}_m$ for emprical distribution with m samples. For NWJ, we calculate the difference

$$|\hat{\mathcal{I}}_m(\boldsymbol{x}, \boldsymbol{y}) - \sup_{\phi \in \mathcal{H}} \hat{\mathcal{I}}[\phi]| \leq \sup_{\phi \in \mathcal{H}} |\mathbb{E}_{\mathbb{P}}[\phi] - \mathbb{E}_{\mathbb{P}_m}[\phi]| + \exp(-1) \sup_{\phi \in \mathcal{H}} |\mathbb{E}_{\mathbb{Q}}[\exp(\phi)] - \mathbb{E}_{\mathbb{Q}_m}[\exp(\phi)]|, \tag{24}$$

where the second term of right-hand side has the same form as equation (32) in Secion 6.2.2 in Belghazi et al. (2018). Therefore, we can follow the proof in Secion 6.2.2 of Belghazi et al. (2018) to prove Lemma 2 for NWJ. Similarly, the similar proving process can be applied to InfoNCE. □

B EXPERIMENTAL DETAILS

All experiments are executed on a single NVIDIA Titan Xp GPU with 12,196M memory.

B.1 TC estimation

Experiment Design Mutual information between two multivariate Guassian distributions $X_1 \sim \mathcal{N}(0, \Sigma_1)$, $X_2 \sim \mathcal{N}(0, \Sigma_2)$ is $\frac{1}{2} \log \frac{\text{Det}(\Sigma_1)\text{Det}(\Sigma_2)}{\text{Det}(\Sigma)}$, where Σ is the covariance matrix of the joint distribution $[X_1, X_2]$.

In our setting, our training samples are sampled from a joint distribution with n variables $[\boldsymbol{X}_i, i \in \{0, 1, 2, 3, n-1\}]$, the marginal distribution of each variable have zero mean and identity covariance matrix with dimension d. Therefore, the determinant of covariance matrix of single variable Σ_i is 1 and the mutual information between two variables i, j is $-\frac{1}{2} \log \text{Det}(\Sigma_{ij})$. The total correlation among variables are $-\frac{1}{2} \log \text{Det}(\Sigma)$. To prove this, we use the idea of line-like structure. Assume that the total correlation of first k variables are $-\frac{1}{2} \log \text{Det}(\Sigma_{[:k]})$, the total correlation of the first $k+1$ variables are

$$-\frac{1}{2} \log \text{Det}(\Sigma_{[:k+1]}) = -\frac{1}{2} \log \text{Det}(\Sigma_{[:k]}) + \frac{1}{2} \log \frac{\text{Det}(\Sigma_{[:k]})}{\text{Det}(\Sigma_{[:k+1]})}, \tag{25}$$

where $\Sigma_{:k}$ is the covariance matrix of the first k variables.

In our proof-of-concept experiments, we set $n = 4, d = 10$. The covariance matrix is

$$\begin{bmatrix} \mathbb{I}_d & \sigma\mathbb{I}_d & 0 & 0 \\ \sigma\mathbb{I}_d & \mathbb{I}_d & 0 & 0 \\ 0 & 0 & \mathbb{I}_d & \sigma\mathbb{I}_d \\ 0 & 0 & \sigma\mathbb{I}_d & \mathbb{I}_d \end{bmatrix}$$

The total correlation under such a design is $-d \log(1 - \sigma^2)$.

As we mentioned in the paper, we can adjust the correlation coefficients (non-diagonal elements) σ to set the ground truth TC values in the set $\{2.0, 4.0, 6.0, 8.0, 10.0\}$.

Hyper-parameters All MI lower bounds require the learning of a value function $f(\boldsymbol{x}, \boldsymbol{y})$; the CLUB upper bound requires the learning of a network approximation $q_\theta(\boldsymbol{y}|\boldsymbol{x})$. To make a fair comparison, we set the value function and the neural approximation with one hidden layer and the same hidden units. For the multivariate Gaussian setup, the number of hidden units is 20. On the top of the hidden layer output, we add the ReLU activation function. The learning rate for all estimators is set to 1×10^{-4}.

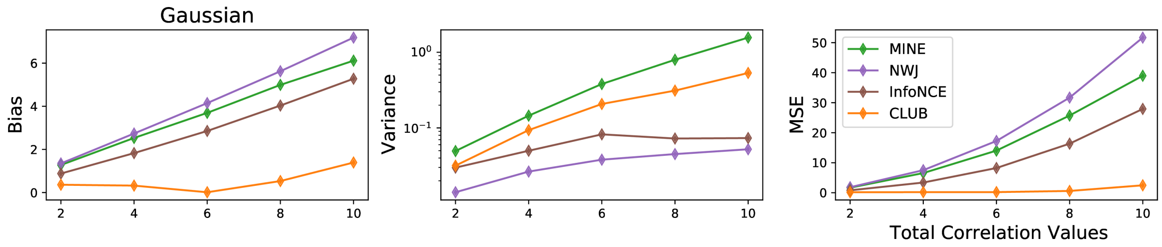


Figure 6: Bias, variance and MSE of **line-like** TC estimators.

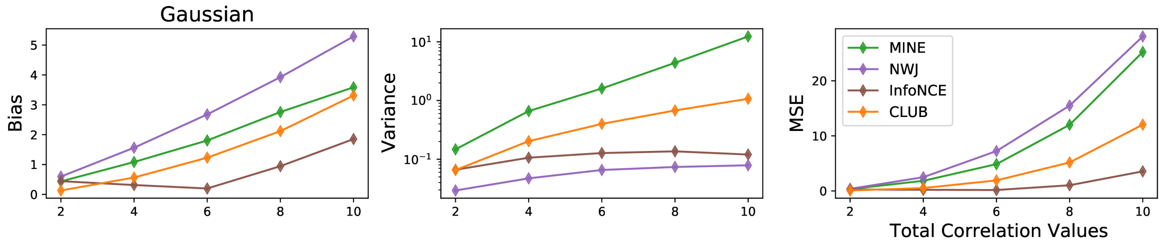


Figure 7: Bias, variance and MSE of **tree-like** TC estimators.

Table 4: Comparison between ours(Tree-based CLUB) with other non-parametric methods under two circumstance with different data dimensionality ($n = 2$ and $n = 10$. The table shows the absolute error between predicted and ground truth. **Bold** means the minimal error and the best predication.

	$n = 2$					$n = 10$				
Total Correlation	2	4	6	8	10	2	4	6	8	10
KNN ((Kraskov et al., 2004b; Pál et al., 2010)	0.08	0.29	0.66	1.29	2.27	1.34	2.66	4.05	5.47	6.98
Bias-improved KNN ((Gao et al., 2018))	0.2	0.59	1.24	2.21	3.54	1.48	2.97	4.51	6.07	7.67
Kernel Density Estimation	1.44	1.40	1.40	1.41	**1.37**	9.02	8.89	8.98	9.00	8.97
Ours	**0.03**	**0.04**	**0.55**	**0.88**	1.63	**0.23**	**0.37**	**0.22**	**1.03**	**1.96**

Bias, Variance, Mean Squared Error Figure 6 7 show the bias, variance and mean squared error using different mutual information estimators. The explanation to this figure is shown in the main paper.

Non-Parametric Methods Since the probability can be directly estimated using non-parametric methods, we can also use non-parametric methods to estimate the total correlation directly. Here we compare with KDE, k-NN based methods (Kraskov et al., 2004b; Pál et al., 2010) and its variant (Gao et al., 2018). As shown table Table 4, the non-parametric methods can only estimate the total correlation decently when the dimension of input n is low. Note that the experiment setting is exactly the same as the main paper except for dimension.

Considering MINE (Belghazi et al., 2018) has already performs better than k-NN based methods (Kraskov et al., 2004b; Pál et al., 2010) and MINE is one of our baselines, our performance should perform better. This is also reflected in the table. Meanwhile, k-Nearest Neighbor estimation method focuses on total correlation estimation only with non-differentiable operation. The estimated value is devoted to variable independence and correlation analysis. While in our neural network-based methods, we calculate the total correlation in a derivative way and get meaningful gradient information for efficient back-propagation. The results are shown in our disentangle and representation learning experiments.

B.2 TC Maximization: Disentanglement

Model Design The dimension of latent space $\boldsymbol{z}, \boldsymbol{z}_{color}, \boldsymbol{z}_{digit}$ is $128, 32, 32$. The structures of each model are shown in Table 5 6 7. We use TC-InfoNCE as our total correlation estimator. Note that when the number of variables is three. There is no difference between tree-based and line-based methods. Both of which require two mutual information estimators, one for mutual information estimation of any pair, the other is to capture the mutual information between the left one and the preselected pair. The parameters of these two estimators are listed in Table 7.

Table 5: Structure of the encoder $\mathcal{E}(\cdot)$, decoder $\mathcal{D}(\cdot)$ described in Section B.2. The terms in brackets of Conv2d and DeConv2d are (input channel, output channel, filter size, stride size, zero padding size, bias included). IN means instance normalization. DeConv2d represents the deconvolution operator.

	Encoder	Decoder
0	Conv2d(3, 64, 4, 2, 1, False), IN, LeakyReLU	DeConv2d(128, 512, 2, 1, 0, True)
1	Conv2d(64, 128, 4, 2, 1, False), IN, LeakyReLU	DeConv2d(512, 256, 4, 2, 1, False), IN, LeakyReLU
2	Conv2d(128, 256, 4, 2, 1, False), IN, LeakyReLU	DeConv2d(256, 128, 4, 2, 1, False), IN, LeakyReLU
3	Conv2d(256, 512, 4, 2, 1, False), IN, LeakyReLU	DeConv2d(128, 64, 4, 2, 1, False), IN, LeakyReLU
4	Conv2d(512, 128, 2, 1, 0, True), ,	DeConv2d(256, 128, 4, 2, 1, False), ,Tanh

Table 6: Structure of classifier $\mathcal{F}_d(\cdot)$ and regressor $\mathcal{F}_c(\cdot)$ described in Section B.2. The terms in the bracket of Linear are (input dimension, output dimension, bias included).

	Color Regressor	Digit Classifier
0	Linear(32, 16, True), ReLU	Linear(32, 16, True), ReLU
1	Linear(16, 3, True), Sigmoid	Linear(16, 10, True),

We compare our method with autoencoder and variational autoencoder. Apart from the reconstruction loss $\mathcal{L}_{recons}$ and KL loss used in AE and VAE, we also take regression loss $\mathcal{L}_{color}$, classification loss $\mathcal{L}_{digit}$ into account. We illustrate the failure transfer case under the AE setup to show the difference without the total correlation term in Figure 8. Both the color and digit are not successfully transferred from the source (bottom row) to the target (rightmost column).

Hyper-parameters The total training epoch and batch size are 300 and 512. Learning rate is $5\mathrm{e}{-3}$ for encoder and decoder, $1\mathrm{e}{-4}$ for the adversarial discriminator, $1\mathrm{e}{-3}$ for the digit classifier and color regressor. All the optimizers are Adam. Since the final loss is the summation of each term, the weight of each term is one except for the color regressor loss, which is ten. We also add spectral normalization (Miyato et al., 2018) to the adversarial discriminator to further stabilize the adversarial training.

B.3 TC Minimization: Sentence Representation

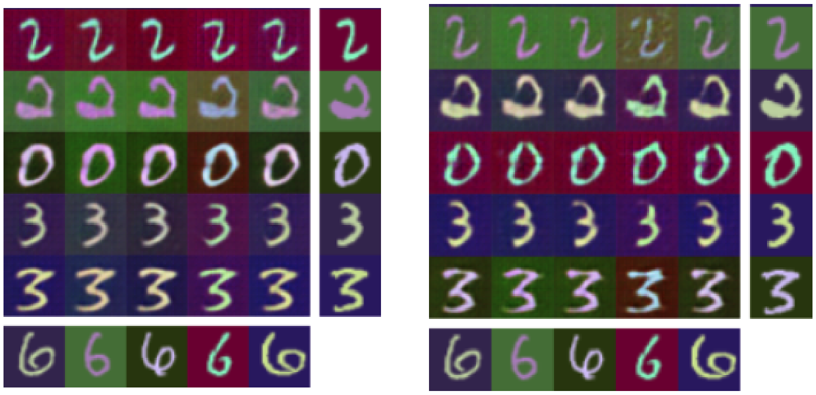
Training Setup We train the model with 10^6 randomly sampled sentences from the English Wikipedia (Foundation) dataset.

Model Structure The model consists of two parts, f is the transformer-based large pretrained language model and g is a multiple layer perception with one hidden layer and ReLU activation function. The hidden neurons and output neurons are 256. The structure of the model follows Gao et al. (2021); Jiang et al. (2022).

Hyper-parameters Considering that f and g are initialized in different ways, we scale up the learning rate of g. In detail, the learning rate and learning rate scale are $1\mathrm{e}{-5}$ and 10 for models with SimCSE-BERT, SimCSE-RoBERTa and PromptBERT as baselines, $5\mathrm{e}{-6}$ and 100 for the model with PromptRoBERTa as baselines. These hyper-parameters are selected according to the validation set. The batch size is 256. The maximum length of a sentence is limited to 32. The training epoch is one. The best model selected based on the performance of the validation set is applied to the testing set. The validation set is evaluated every 250 steps.

Augmentations Following the simple but effective data augmentation techniques in SimCSE, we get four augmentations by feeding the same input to the network four times and sample four dropout samples independently. Prompt-BERT also applies this dropout strategy, while they modify the inputs as well as the output feature representation neuron.

SimCSE and PromptBERT both apply the InfoNCE with cosine similarity as the score function and use the large masked language pretrained model like BERT(Devlin et al., 2018) as the initialization of the feature encoder $f(\cdot)$. The difference is how they construct the paired distribution $p(x, y)$. SimCSE feeds the same sentence to the encoder twice and obtains different output embeddings by independently sampling two dropout masks of the encoder $f(\cdot)$ ([CLS] representation is taken as the representation of the sentence). PromptBERT adapts two fixed cloze-style templates to augment the original



(a) color transfer

(b) digit transfer

Figure 8: Failure case of disentangle transfer, using autoencoder without the total correlation constraint. The bottom row is the source image and rightmost column is the target image.

Table 7: Structure of the adversarial discriminator $\mathcal{H}(\cdot)$ and two total correlation estimators described in Section B.2. The terms in brackets of Conv2d and DeConv2d are (input channel, output channel, filter size, stride size, zero padding size, bias included). The terms in the bracket of Linear are (input dimension, output dimension, bias included). BN means batch normalization.

	Discriminator	TC estimator 1	TC estimator 2
0	Conv2d(3, 32, 5, 1, 2, False), BN, ReLU	Linear(128, 90, True), ReLU	Linear(64, 64, True), ReLU
1	Conv2d(32, 64, 5, 1, 2, False), BN, ReLU	Linear(90, 90, True), ReLU	Linear(64, 64, True), ReLU
2	Linear(4096, 1), ,	Linear(90, 1, True), Softplus	Linear(64, 1, True), Softplus

Table 8: Four templates used in PromptBERT-$\mathcal{TC}$ and PromptRoBERTa-$\mathcal{TC}$. [CLS],[MASK], [SEP] are special tokens used in BERT. [X] is the placeholder of the input sentence.

	BERT	RoBERTa
Template 0	[CLS] This sentence of "[X]" means[MASK].[SEP]	[CLS] This sentence of ’[X]’ means[MASK].[SEP]
Template 1	[CLS] The sentence : "[X]" means [MASK].[SEP]	[CLS] The sentence : ’[X]’ means [MASK].[SEP]
Template 2	[CLS] The sentence ’ [X] ’ has the same meaning with [MASK].[SEP]	[CLS] The sentence ’ [X] ’ has the same meaning with [MASK].[SEP]
Template 3	[CLS] [MASK] has similar meaning with sentence : ’[X]’.[SEP]	[CLS] [MASK] has similar meaning with sentence : ’[X]’.[SEP]

sentence. For example, a template could be “this sentence : [X] means [MASK].”, where [X] is the placeholder of the origin input sentence and [MASK] is a special placeholder token. The representation of [MASK] token $f(X)_{\text{[MASK]}}$ is donated as the sentence embedding.

In our total correlation estimation experiments, we augment the input sentence four times. More specifically, we sample four drop-outs for SimCSE and design four cloze-style templates for Prompt-BERT (Table 8). Our design of the template partially follows Jiang et al. (2022).