Ph.D. Dissertation of Engineering

Understanding Mutual Information in Contrastive Representation Learning

대조 표현 학습에서 상호 정보의 이해

February 2023

Program in
Digital Contents and Information
Graduate School of
Convergence Science and Technology
Seoul National University

Kyungeun Lee

Understanding Mutual Information in Contrastive Representation Learning

Advisor Wonjong Rhee

Submitting a Ph.D. Dissertation of Engineering

February 2023

Program in Digital Contents and Information
Graduate School of
Convergence Science and Technology
Seoul National University

Kyungeun Lee

Confirming the Ph.D. Dissertation written by
Kyungeun Lee

February 2023

Chair	Nojun Kwak	(Seal)
Vice Chair	Wonjong Rhee	(Seal)
Examiner	Kyogu Lee	(Seal)
Examiner	Bongwon Suh	(Seal)
Examiner	Daeyoung Choi	(Seal)

Abstract

Contrastive learning has played a pivotal role in the recent success of unsupervised representation learning. It has been commonly explained with instance discrimination and a mutual information loss, and some of the fundamental explanations are based on mutual information analysis. An analysis based on mutual information, however, can be misleading. First of all, an exact quantification of mutual information over a real-world dataset is challenging. It has not been solved because we cannot access the true joint distribution function of real-world dataset before. Second, previous studies have equated the limitations of contrastive learning with them of mutual information estimation in the absence of the rigorous investigation for a relationship between them. Third, what information is actually being shared by the two views is overlooked. Without carefully examining what information is actually being shared, the interpretation can be completely misleading. In this work, we develop new methods that enable rigorous analysis of mutual information in contrastive learning. We also evaluate the accuracy of variational MI estimators across various data domains, including images and texts. Using the methods, we investigate three existing beliefs and show that they are *incorrect*. Based on the investigation results, we address two issues in the discussion section. In particular, we question if contrastive learning is indeed an *unsupervised* representation learning method because the current framework of contrastive learning relies on validation performance for tuning the augmentation design.

Keywords: representation learning, information theory, mutual information, variational bounds of mutual information, deep representations, contrastive learning

Student number: 2016-23985

Table of Contents

Chapter 1. Introduction **1**

1.1 Contributions 6

Chapter 2. Background **9**

2.1 Contrastive representation learning 9

2.1.1 Previous works to understand contrastive learning 11

2.2 Mutual Information 12

2.3 Variational Mutual Information Estimators 15

2.3.1 Critic function 18

2.3.2 Limitations of the variational MI estimators 19

Chapter 3. Same-class Sampling for Positive Pairing **21**

Chapter 4. Understanding the Accuracy of Variational Mutual Information Estimators **27**

4.1 Datasets 29

4.1.1 Gaussian dataset 30

4.1.2 Definitions of d_s, d_r, and Z 30

4.1.3 Details of generating datasets 31

4.2 Experimental setup 33

4.3 Experimental results 34

4.3.1 Critic architecture 34

4.3.2 Critic capacity 38

4.3.3 Choice of the variational MI estimator 39

4.3.4 Number of information sources (d_s) 39

4.3.5 Representation dimension (d_r) 40

4.3.6 Nuisance 41

4.3.7 Deep representations 41

4.4 Discussion: How can we make use of MI with practical datasets? . . . 44

4.5 Conclusion 48

Chapter 5. Examining Three Existing Beliefs on Mutual Information in Contrastive Learning 49

5.1 Method 50

5.1.1 Post-training MI estimation 50

5.1.2 CDP dataset 52

5.2 Experimental setups 56

5.2.1 Training 56

5.2.2 Post-training MI estimation 57

5.3 Results 59

5.3.1 A small batch size is a limiting factor for MI estimation but not for contrastive learning. 59

5.3.2 Augmentation-based MI and other metrics are not effective, but $\hat{I}_{\text{class}}(h_X; h_Y)$ is effective. 62

5.3.3 Minimizing task-irrelevant information (InfoMin) is not always necessary. 70

5.4 Discussion 77

5.5 Conclusion 83

Chapter 6. Conclusion 84

6.1 Limitations 86

6.2 Future works 86

Bibliography 87

Appendices 98

A Implementation details of Chapter 5 99

A.1 SimCLR augmentation ($\mathcal{T}_{\text{SimCLR}}$) 99

A.2 Full results of Table 5.3 101

Acknowledgement 106

List of Tables

2.1 Summary of variational mutual information estimators. For the optimization step, we find $f^*(x, y)$ to maximize the optimization loss $\mathcal{L}(f(x, y))$ for a given batch size K. For the estimation step, we evaluate the MI values with the variational bounds. 18

5.1 Post-training MI estimation results for ResNet-50 on ImageNet-100 and ImageNet-1k. Sixteen pre-trained models are used to evaluate the effectiveness of $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ and $\hat{I}_{\text{class}}(h_X; h_Y)$. 66

5.2 Post-training MI estimation results for ViT on ImageNet-100 and ImageNet-1k. Fourteen pre-trained models are used to evaluate the effectiveness of $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ and $\hat{I}_{\text{class}}(h_X; h_Y)$. Because of the computational budge, we exclude the two largest models for ImageNet-1k. 67

5.3 Summary of seven experiments. Except for $\hat{I}_{\text{class}}(h_X; h_Y)$ that directly utilizes downstream class information in $\mathcal{T}_{\text{class}}$, all the other known metrics turn out to be ineffective for assessing downstream task performance. In the case of alignment and uniformity, smaller values indicate better representations, so we flipped the signs. Note that the class label information is also utilized by tolerance and linear CKA. 68

5.4 Theoretical upper bounds of the supervised loss for CDP dataset. All the bounds are determined based on the same variables, including the batch size, the number of class, and the contrastive loss. Since we fix the batch size and the number of classes, only the contrastive loss affects the bounds. Thus, all bounds have the same correlation coefficient of $\rho = -0.409$ and $\tau_K = -0.182$. We follow the official implementation codes of (Bao, Nagano, and Nozawa 2022). 69

5.5 Comparison of linear evaluation performance for a set of loss functions. Performance with $*$ indicates values reported in the existing works. Despite the superiority of $\mathcal{L}_{\text{class}}$ as a metric, generally it does not outperform the best known unsupervised losses. 79

5.6 The effect of negative sampling dataset $\mathcal{D}^-$. Linear evaluation performance can be affected by choosing negative samples from a related or an unrelated dataset. (a) CIFAR-5A: For contrastive learning of CIFAR-5A dataset, the best performance is achieved by choosing the negative samples from CIFAR-5B dataset (i.e., not from CIFAR-5A dataset). (b) CIFAR-50A: For contrastive learning of CIFAR-50A dataset, the best performance is achieved by choosing the negative samples from CIFAR-50B dataset (i.e., not from CIFAR-50A dataset). 81

A.1 Metrics evaluated on CDP dataset. 101

A.2 Metrics evaluated on CIFAR-10 dataset. 102

A.3 Metrics evaluated on ImageNet-100 dataset. 102

A.4 Metrics evaluated on ImageNet-100 dataset using pre-trained ResNet-50 models. 103

A.5 Metrics evaluated on ImageNet-1k dataset using pre-trained ResNet-50 models. 104

A.6 Metrics evaluated on ImageNet-100 dataset using pre-trained ViT models. 105

A.7 Metrics evaluated on ImageNet-1k dataset using pre-trained ViT models. Because of the computational budget, we exclude the two largest models. 105

List of Figures

1.1 Illustration of the motivation of this study. To explain the success of contrastive learning, some key factors are frequently referenced, e.g., instance discrimination and aggressive augmentation. Among the numerous key factors, we focus on InfoNCE loss, which has a strong analogy with MI. To relate the success of contrastive learning to maximizing MI, we require rigorous investigation. Note that we define good representation as a representation with high downstream-task performance for the purpose of study. 4

1.2 A simple example whereby the estimated MI $\hat{I}(h_X; h_Y)$ and true MI $I(h_X; h_Y)$ exhibit a clearly different relationship with the downstream-task accuracy. 5

1.3 The choice of augmentation method affects the joint distribution $p(x, y)$, and the joint distribution $p(x, y)$ affects the shared information $I(X; Y)$. Thus, the choice of augmentation method is critical when we evaluate the MI between two views. 6

2.1 Illustration of contrastive representation learning for an image dataset. InfoNCE loss has been a general choice for the loss function to pull together positive pairs (two views generated from the same image) and to push apart negative pairs (the other views in the same batch) simultaneously. [1] 9

2.2 Illustration of the relationship between entropy and mutual information. (The figure was adapted from Figure 2.2 of (Cover 1999).) 14

3.1 Positive pairing method implicitly determines the joint and marginal distributions – $p(x, y)$ and $p(x)p(y)$ are determined by the choice of $\mathcal{T}$. 21

[1] (Figure is adapted from `https://blog.salesforceairesearch.com/`.)

3.2 Illustration of same-class sampling. (a) The black solid arrows refer to same-class sampling. We sample the positive pair (x_i, y_i) to share the information source c_i and no additional information. The blue dotted arrows refer to an assumption of Theorem 2 whereby an error-free classification function $h_{\text{class}} : X \to C$ exists. (b) An example of ImageNet. 22

3.3 Markov process of same-class-sampling, $\mathcal{T}_{\text{class}}$. S denotes the anchor image, C denotes the image's downstream task class label, and X and Y correspond to the positive pair chosen for same-class-sampling of the image S. (a) The original Markov process of same-class-sampling. (b) Equivalent Markov process of the same-class-sampling. 23

3.4 While we have no access to the true MI value in the case of the general augmentation methods, we can make use of true MI for any type of dataset in the case of same-class sampling. Even though the assumption is not guaranteed, we have access to the upper bound of true MI for any dataset. 26

4.1 Construction of image dataset with non-integer MI value by utilizing the binary symmetric channel. 32

4.2 Example of generating a dataset with known true MI value for images. X and Y consist of random images drawn from the MNIST dataset. (a) Basic construction: the digits appear together with probability 0.5 and $I(X;Y) = 1$. (b) Concatenate four samples in (x, y)-coordinates: four samples have independent labels and $I(X;Y) = 4$. (c) Concatenate three samples in channel dimension: three samples have independent labels and $I(X;Y) = 3$. (d) Adding nuisance: an independently chosen background image from CIFAR-10 is inserted as the nuisance. Nuisance does not affect the true MI; therefore, $I(X;Y) = 1$. 33

4.3 Estimation results for three different datasets. We found that the variational estimators of mutual information are surprisingly accurate for image and sentence embeddings, better than Gaussian datasets. . . . 36

4.4 Image and sentence embeddings, $d_s = 10$, $d_r = 7680$ 37

4.5 Mean squared error for the different depths of a joint critic function when true MI is 2 bits. For all datasets and estimators, we found no distinguishable improvements when increasing the depth of the critic f_{joint}. 38

4.6 Estimation results for three different datasets for different ground-truth MI values. While the SMILE estimator provides a clear improvement for the Gaussian dataset, the MINE and NWJ estimators provide better estimation accuracy for the image dataset. 39

4.7 Estimation fails when d_s is large. Shades correspond to the standard deviation of the estimations. For all datasets and estimators, the estimation fails when d_s is too large. 40

4.8 MI estimation results when representation dimension d_r is increased. . 40

4.9 Estimation fails when too severe a nuisance is inserted. Although the nuisance does not affect the true MI value, the estimation fails when η is greater than 0.4. 41

4.10 Estimation results for deep representations. For all types of encoder networks, we achieve accurate estimations regardless of the choice of network architecture. 43

4.11 MI estimation results of the image dataset when a nuisance is inserted with $\eta = 1$. While we cannot achieve tight bounds for raw inputs, we achieve accurate estimations after the encoder network is trained for all types of estimators and all values of true MI. 45

4.12 MI estimation results for hidden layers of ResNet-50. Dashed lines indicate boundaries between stages. We found that the upper layer representations retain the task-relevant information C. The transition clearly occurs when the output size changes. Further, the top layers provide the tight bound of the true MI while the lower layers do not. . 46

4.13 MI estimation results for ImageNet-100 dataset. We test 30 pre-trained models (16 for ResNet-50 and 14 for ViT). When the linear accuracy is close to 100%, we can achieve accurate estimation of the true MI. . . 47

5.1 Training and MI estimation. (Top) Training: We train the encoder $f_e(\cdot)$ and the projection head $f_p(\cdot)$ to minimize the InfoNCE loss $\mathcal{L}$. (a) With unsupervised positive pairing $\mathcal{T}_{\text{SimCLR}}$. (b) With supervised positive pairing $\mathcal{T}_{\text{class}}$. (Bottom) Post-training MI estimation: We train the critic $f_c(\cdot)$ to maximize the InfoNCE bound $\hat{I}(h_X; h_Y)$ while $f_e(\cdot)$ is frozen. (c) With unsupervised positive pairing $\mathcal{T}_{\text{SimCLR}}$. (d) With supervised positive pairing $\mathcal{T}_{\text{class}}$. 51

5.2 Manipulating true MI with CDP dataset. When only one of color, digit, and position is matched by same-class sampling as shown in (a), (b), and (c), the true MI is 2 bits ($I_{\text{class}}(X;Y) = 2$). When all three are consistently matched as shown in (d), the true MI is 6 bits ($I_{\text{class}}(X;Y) = 6$). 52

5.3 An example of how to generate a CDP dataset. Detailed descriptions are provided in the text. 53

5.4 An example of applying same-class sampling for the CDP dataset and the practical dataset (ImageNet-1k). 55

5.5 An example of same-class sampling for CDP dataset. 56

5.6 Examples of post-training MI estimation: (a) CDP, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (b) CDP, $\hat{I}_{\text{class}}(h_X; h_Y)$, (c) CIFAR-10, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (d) CIFAR-10, $\hat{I}_{\text{class}}(h_X; h_Y)$, (e) ImageNet-100, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (f) ImageNet-100, $\hat{I}_{\text{class}}(h_X; h_Y)$, (g) ImageNet-1k, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (h) ImageNet-1k, $\hat{I}_{\text{class}}(h_X; h_Y)$. Note that the MI estimation $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ in (a) is relatively smaller when compared to the $\hat{I}_{\text{class}}(h_X; h_Y)$ in (b). This is an example where $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ does not properly reflect the downstream task performance while $\hat{I}_{\text{class}}(h_X; h_Y)$ does. 58

5.7 A summary of experimental setups. For both phases, we use same-class sampling for positive pairing with CDP dataset. For training, we utilize the batch size K_{Tr} from 2 to 256. For post-training MI estimation, we fix the batch size K_{Est} as 256. 60

5.8 If the existing belief is correct, the results should be as above. (Left) The downstream-task accuracy should be improved when we increase the training batch size. (Right) We cannot estimate the MI larger than $\mathcal{O}(\log K)$. 61

5.9 True MI, training MI (conventional estimating of MI at the time of training using the training loss), and our post-training MI. For CDP dataset, we train two ResNet models using $\hat{I}_{\text{class}}(h_X; h_Y)$ as the loss (as in Figure 5.1(b)). We evaluate $\hat{I}_{\text{class}}(h_X; h_Y)$ at the end of training (blue) and post-training (orange). During training, the MI is upper bounded by $\log(2K_{\text{Tr}} - 1)$ (dashed lines of green color). After the training is complete, the network is frozen and we evaluate the MI using a large batch size of $K_{\text{Est}} = 256$. Even though the training MI is limited by the $\log(2K_{\text{Tr}} - 1)$ bound, the post-training MI turns out to be almost the same as the true MI ($=$ 6 bits). Obviously, the trained model can represent sufficiently large amount of information. 61

5.10 A summary of experimental setups. To examine the good representations has the optimal metric, we investigate six representation metrics. We do not need to make use of the exact value of true MI, so we use various practical datasets, including CIFAR-10 and ImageNet. Also, we do not need to consider how the encoder network is trained, so we use lots of various pre-trained models. Then, we evaluate each metric in post-training phase. 65

5.11 A description of InfoMin principle. The figure is adapted from Figure 1 of (Tian et al. 2020). 71

5.12 An example of InfoMin principle with CDP dataset. 74

5.13 A summary of experimental setups. We first train the encoder network based on same-calss sampling for positive pairing. Then, we evaluate the learned representations for individual classification tasks. 75

5.14 Linear evaluation performance of CDP dataset for task-dependent training. The task in x-axis indicates the positive pairing $\mathcal{T}$ used for training. We choose one of $\left\{\mathcal{T}_{\text{class}}^{\text{color}}, \mathcal{T}_{\text{class}}^{\text{digit}}, \mathcal{T}_{\text{class}}^{\text{position}}, \mathcal{T}_{\text{class}}^{\text{all}}\right\}$. The task in y-axis indicates the evaluated downstream task C. We choose one of $\left\{C_{\text{color}}, C_{\text{digit}}, C_{\text{position}}, C_{\text{all}}\right\}$. 75

5.15 Post-training MI estimation results for the experiment cases in Figure 5.14. 75

5.16 Linear evaluation performance when $\mathcal{T}_{\text{aug}}^{\text{ColorJitter}}$ is used for training. We have tuned the strength of color jittering in a way similar to (Tian et al. 2020). While we have found a similar result for the downstream task of color classification, the peak in the middle was found also for the other three tasks. Considering that color jittering is not related to digit task or position task, our result indicates that the peak in the middle might not be relevant to InfoMin. 76

5.17 Linear evaluation performance when $\mathcal{T}_{\text{aug}}^{\text{RandomResizedCrop}}$ is used for training. We have tuned the minimum scale parameter of random resized crop in a way similar to (Tian et al. 2020). 76

Chapter 1. Introduction

Among the core concepts of information theory, Kullback–Leibler (KL) divergence has been a popular ingredient for machine learning. KL divergence serves as a dissimilarity measure between two probability distributions. Another core concept of information theory is Mutual Information (MI). MI quantifies the shared Shannon information between two random variables, and it can serve as a fundamental measure of dependency (Cover 1999). MI has become increasingly popular in recent deep learning studies, including generative models (Chen et al. 2016) and language representation learning (Oord, Li, and Vinyals 2018; Wang et al. 2020). In particular, (Hjelm et al. 2018; Bachman, Hjelm, and Buchwalter 2019; Chen et al. 2020a; Chen and He 2020; Grill et al. 2020a) studied self-supervised learning and achieved promising results using contrastive loss (Oord, Li, and Vinyals 2018; Chen et al. 2018), where contrastive loss has a strong analogy to the mutual information.

While MI is an elegant concept with extensive use cases, its exact evaluation over real-world datasets is almost always impossible because of the unknown $p(x, y)$. Even when the joint probability distribution is known, evaluation can be challenging because the integration over $p(x, y)$ is nontrivial. To account for these problems, variational MI estimators have been developed (Belghazi et al. 2018; Poole et al. 2019; Song and Ermon 2019; 2020; Cheng et al. 2020). The variational estimators are based on two steps. First, an analytical bound is derived where the bound is based on a critic function $f(x, y)$. Second, the bound is made tight by optimizing for a supremum or an infimum over $f(x, y)$. In recent works, deep neural networks have been used to model the critic function. When a proper loss function is chosen and the learning of $f(x, y)$ is successful, the variational estimations have been shown to be accurate for a toy dataset.

To assess an MI estimator, a Gaussian dataset based on a multivariate Gaussian model has been used (Belghazi et al. 2018; Poole et al. 2019; Song and Ermon 2019;

2020; Cheng et al. 2020). The toy dataset is convenient because it offers a simple analytical formula of the true MI, $I(X;Y)$. With the true MI, the characteristics of the estimated MI, $\hat{I}(X;Y)$, can be assessed. For instance, accuracy, bias, and variance of variational MI estimators have been analyzed by comparing $I(X;Y)$ and $\hat{I}(X;Y)$ (Belghazi et al. 2018; Poole et al. 2019; Song and Ermon 2019; 2020; Cheng et al. 2020). The Gaussian dataset, however, is far from being representative of real-world datasets, because its underlying structure is purely statistical. For real-world datasets with strong manifold structures, it has been pointed out that variational MI estimators might be inaccurate and misleading (Song and Ermon 2019). It would be ideal if real-world datasets with known true MI were available. In the absence of such datasets, (Song and Ermon 2019) proposed three types of self-consistency test and found that none of the existing MI estimators passed all the test. The result, however, is limited in that a direct assessment of estimator accuracy is missing. To address this issue, we propose a same-class sampling for positive pairing (Chapter 3). The idea is to pair two inputs to have the same class label such that the two inputs' true MI is determined at the time of positive pairing. With the same-class sampling, we can assess the true MI for any type of dataset under a mild assumption. In fact, we can control the true MI. To the best of our knowledge, this is the first trial to use a non-Gaussian dataset with known true MI values.

Even though we could assess the true MI of non-Gaussian datasets, we still have an obstacle to using the variational MI estimators for analyzing the deep representations. We need to investigate how MI estimators work for various data domains. In Chapter 4, we conduct various experiments to scrutinize the estimation accuracy of variational approaches on three different data domains, namely multivariate Gaussian, images, and sentence embeddings. After we verify the accuracy of variational MI estimators, we can examine the role of mutual information in unsupervised representation learning by estimating MI between representations. In fact, MI has many use-cases in deep learning applications, including non-vacuous generalization bounds (Xu and Raginsky

2017), understanding training dynamics (Tishby and Zaslavsky 2015; Achille and Soatto 2018), censoring representations (Moyer et al. 2018), and representation learning based on the InfoMax principle (Bell and Sejnowski 1995; Hyvärinen and Oja 2000; Oord, Li, and Vinyals 2018; Hjelm et al. 2018). In this study, we focus on contrastive learning (Oord, Li, and Vinyals 2018), one of the most successful approaches in unsupervised representation learning.

Contrastive learning has achieved remarkable success in the field of unsupervised representation learning (Oord, Li, and Vinyals 2018; Belghazi et al. 2018; Bachman, Hjelm, and Buchwalter 2019; Henaff 2020; Chen et al. 2020b; Tian, Krishnan, and Isola 2020; He et al. 2020; Khosla et al. 2020; Chen et al. 2020c; Gao, Yao, and Chen 2021; Xie et al. 2021; Sordoni et al. 2021; Purushwalkam and Gupta 2020; Wu et al. 2020a; Mitrovic et al. 2020; Tsai et al. 2020). When there are no annotations, contrastive learning generates multiple views of a given image and learns useful representations by pursuing an invariance. For this pretext task of instance discrimination, it has been empirically found that *InfoNCE loss* (Oord, Li, and Vinyals 2018; Gutmann and Hyvärinen 2010) is an effective training objective for a variety of downstream tasks. InfoNCE loss not only plays a key role in achieving a robust and outstanding performance, but it also provides an elegant interpretation where the representation learning can be understood as a *Mutual Information* (MI) maximization between the two augmented views (X and Y) of a given image (Oord, Li, and Vinyals 2018; Hjelm et al. 2018; Bachman, Hjelm, and Buchwalter 2019; Tian, Krishnan, and Isola 2020; Sordoni et al. 2021). Numerous works have studied contrastive learning based on the theoretical interpretation, and some have become fundamental and crucial for understanding contrastive learning.

An analysis based on MI of X and Y, however, can be tricky and misleading. We summarize the limitations of previous approaches as follow:

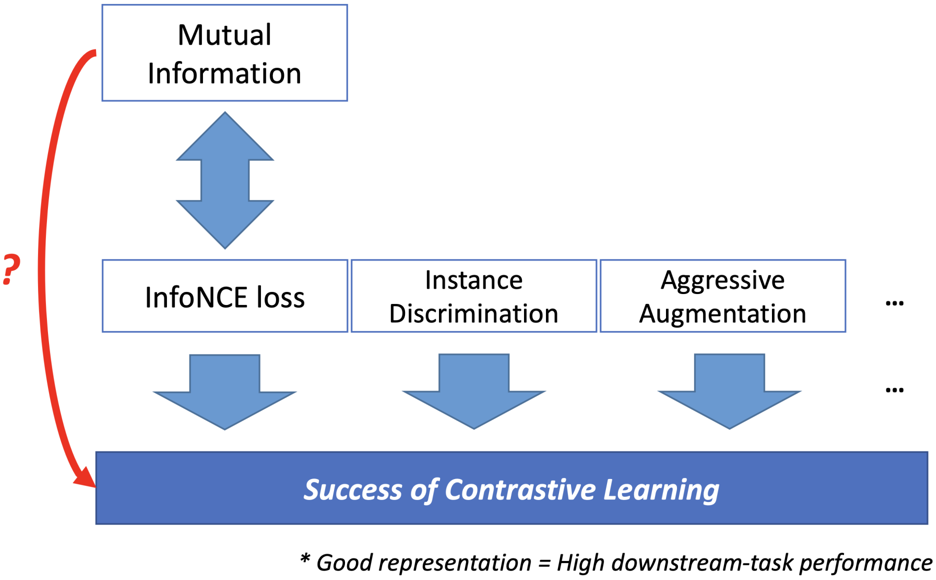

Mutual Information
?
InfoNCE loss
Instance Discrimination
Aggressive Augmentation
...
...
Success of Contrastive Learning
* Good representation = High downstream-task performance

Figure 1.1. Illustration of the motivation of this study. To explain the success of contrastive learning, some key factors are frequently referenced, e.g., instance discrimination and aggressive augmentation. Among the numerous key factors, we focus on InfoNCE loss, which has a strong analogy with MI. To relate the success of contrastive learning to maximizing MI, we require rigorous investigation. Note that we define good representation as a representation with high downstream-task performance for the purpose of study.

Can we believe the estimation values? First, precaution is needed to interpret the estimated MI value because the variational MI estimator provides only the lower bound of MI. We explicate using a toy example as shown in Figure 1.2. If we simply regard the estimated MI (orange line) as equivalent to the true MI (gray line), we might derive an invalid conclusion that maximizing MI is beneficial. Because the estimated MI is only the lower bound of the true MI, we must utilize the true MI values to evaluate the estimated values. Second, an exact evaluation of MI requires the joint distribution function $p(x, y)$, but $p(x, y)$ is not directly accessible for practical problems. For practical problems with complex neural representations and intractable $p(x, y)$, the neural estimators based on variational bounds are known to be the most reliable (Belghazi et al. 2018; Poole et al. 2019). The neural estimators, however, do not guarantee a sound analysis because they can only provide estimates, and we cannot tell if the estimates are sufficiently accurate for the analysis of interest. This problem can be alleviated with a practical dataset with known true MI values, but many previous works simply assumed the estimates to be exact in the absence of true MI values.

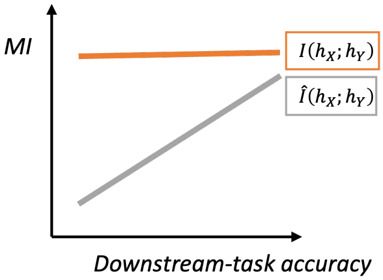


Figure 1.2. A simple example whereby the estimated MI $\hat{I}(h_X; h_Y)$ and true MI $I(h_X; h_Y)$ exhibit a clearly different relationship with the downstream-task accuracy.

What information is being shared by the two views? Is this what we are interested in? The choice of data augmentation determines the joint distribution function $p(x, y)$, and $p(x, y)$ determines not only the shared information $I(X; Y)$ (and $I(h_X; h_Y)$) but also what will be learned during training. Without carefully examining the information actually being shared by the two views, the interpretation can be completely misleading. This issue can be alleviated by introducing the specific type of augmentation to limit the shared information to be task-relevant information only.

Can we regard the limitations for MI estimation equivalent to the limitations for contrastive learning? The limitations of the MI estimators should be carefully related to the limitations of what contrastive learning can learn. Because of the use of InfoNCE loss as the objective of contrastive learning, where InfoNCE is also a popular

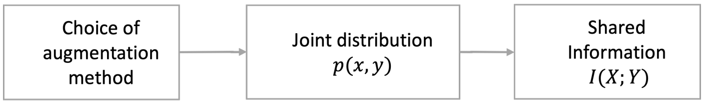

Choice of augmentation method
Joint distribution $p(x, y)$
Shared Information $I(X; Y)$

Figure 1.3. The choice of augmentation method affects the joint distribution $p(x, y)$, and the joint distribution $p(x, y)$ affects the shared information $I(X; Y)$. Thus, the choice of augmentation method is critical when we evaluate the MI between two views.

MI estimator (Oord, Li, and Vinyals 2018; Poole et al. 2019; Song and Ermon 2019; Tschannen et al. 2019), many previous works incorrectly assumed the limitations to be the same for both MI estimation and contrastive learning.

In Chapter 5, we examine the role of mutual information in contrastive learning based on the carefully designed experiments. MI has often been referred to as the important factor for designing and analyzing contrastive learning. In existing works, however, most naively used MI based on arbitrary augmentation and instance discrimination for the analysis. For example, (Tschannen et al. 2019) defined two views as the top and bottom parts of an MNIST image, and they estimated the MI between them. As a result, they concluded that MI does not relate well with the downstream task performance. However, this is due to the particular choice of augmentation and the resulting MI and should not be considered as a general property of MI.

1.1. Contributions

In this study, we develop a set of rigorous methods for analyzing MI in contrastive learning and show that the following three existing beliefs should be reconsidered.

1. A small batch size is undesirable for contrastive learning because of InfoNCE's $\mathcal{O}(\log K)$ bound (Hjelm et al. 2018; Tian, Krishnan, and Isola 2020; Bachman, Hjelm, and Buchwalter 2019; Wu et al. 2020a; Song and Ermon 2020; Chen et al. 2020b; Sordoni et al. 2021).

2. MI cannot measure how effective the representation is for the downstream task's

performance (Tschannen et al. 2019). Instead, other metrics, such as uniformity (Wang and Isola 2020; Wang and Liu 2021), alignment (Wang and Isola 2020), tolerance (Wang and Liu 2021), and linear CKA (Nguyen, Raghu, and Kornblith 2020; Song et al. 2012; Nguyen, Raghu, and Kornblith 2022), are more relevant and useful than MI.

3. To design optimal views, task-irrelevant information must be discarded for a better generalization (Tian et al. 2020; Tsai et al. 2020; Xiao et al. 2020; Chen, Luo, and Li 2021).

For a rigorous investigation, in Chapter 3, we develop an analysis framework based on three key elements. First, we clarify that *the choice of augmentation design* dictates the shared information between the two views. While this may sound obvious, it is a crucial step for cautiously investigating contrastive learning, because the choice of augmentation design directly commands the joint distribution $p(x, y)$; consequently, $p(x, y)$ decides the MI of learning, and ultimately the MI determines what will be learned as the representation. A specific choice of augmentation, named *same-class sampling* in our work, plays a pivotal role in our study. This is special because it only shares class information between the two views and its true MI can be proven to be the same as the class entropy $H(C)$ under a mild assumption. Second, we use a dedicated phase of MI estimation called *post-training MI estimation*. In previous works, MI estimation was typically performed concurrently during the training phase, because the InfoNCE can be conveniently used not only as the training loss but also as the variational estimator. Separating MI estimation into a post-training phase allows us to compare a wide scope of representation encoders because it is applicable to any representation encoder (e.g., a basic supervised network learned with the cross-entropy loss). Third, we introduce the *CDP dataset* that allows information to be embedded in an image by varying color, digit, and position. Due to the way the CDP dataset is constructed, the true MI value can be easily manipulated by controlling the dependency among the three attributes over the two views. Using the CDP dataset, we were able to construct a few experiments without any ambiguity in interpretation. In addition, we were able to confirm that the MI estimation values in our experiments are accurate. This was made possible by comparing the theoretically derived true MI values with the estimated MI values.

1. A small batch size limits the training loss, but it limits neither the information in the learned representation nor the downstream-task performance.

2. The only metric (among the metrics that we have investigated) that is strongly relevant to the downstream-task performance is the MI of the downstream-task information itself.

3. Task-irrelevant information does not necessarily harm the generalization of the downstream task.

Finally, we discuss two essential issues based on our investigation results. First, we clarify that a properly chosen MI is an excellent metric for evaluating representations. However, the same metric is not an effective training objective for successful representation learning. Second, we raise the question of whether contrastive learning is really an unsupervised representation learning method. The current framework heavily relies on a heuristic and extensive tuning of the augmentation design based on a validation dataset. Apparently, it still remains open to developing a further advanced representation learning framework compared to contrastive learning.

Chapter 2. Background

In this chapter, we provide the background information related to contrastive representation learning, mutual information, and variational approaches to estimate MI. We first introduce contrastive representation learning based on image cases. To clearly identify the relationship between contrastive loss and MI, we provide detailed derivations, and we show how minimizing contrastive loss is equivalent to maximizing MI. Then, we provide a brief introduction for mutual information. Finally, we summarize variational MI estimators and the limitations of them.

2.1. Contrastive representation learning

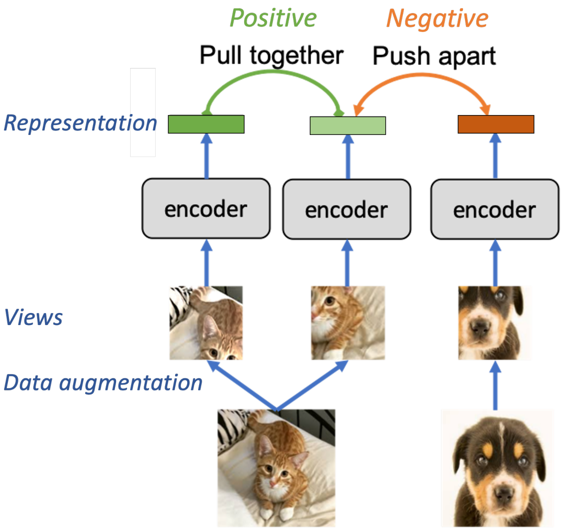


Figure 2.1. Illustration of contrastive representation learning for an image dataset. InfoNCE loss has been a general choice for the loss function to pull together positive pairs (two views generated from the same image) and to push apart negative pairs (the other views in the same batch) simultaneously. [1]

Given a dataset $\mathcal{D} = \left\{ s_i | s_i \in \mathbb{R}^m \right\}$, we can sample an image s_i, generate its views with a family of augmentations $\mathcal{T}$, and randomly select two of them to form a positive pair (x_i, y_i). See Figure 3.1(b) for an example, where $\mathcal{T}$ is a family of SimCLR (Chen et al. 2020b) augmentations. After repeating this K times, InfoNCE loss for a batch can be calculated as

$$\mathcal{L} = \frac{1}{2K} \sum_{k=1}^{K} [l(2k-1, 2k) + l(2k, 2k-1)] \tag{2.1}$$

$$\text{with} \;\; l(i,j) = -\log \frac{\exp\left(z_{i,i}/\tau\right)}{\sum_{k=1}^{2K} \mathbb{1}_{[k \neq i]} \exp\left(z_{i,j}/\tau\right)}, \tag{2.2}$$

where $z_{i,j} = \text{sim}(f(x_i), f(y_j))$; $f = f_p \circ f_e$ with $f_e(\cdot)$ as the encoder and $f_p(\cdot)$ as the projection head; $\text{sim}(u, v) = u^T v / ||u|| ||v||$ denotes the dot product between l_2 normalized u and v (i.e. cosine similarity); τ denotes a temperature scalar; and K is the batch size. We denote the encoded representation vector of an input X as $h_X = f_e(X)$. While the InfoNCE loss can be used for training, it can be slightly modified to the following InfoNCE bound and used for MI estimation as well.

$$\hat{I}(h_X; h_Y) = \log(2K - 1) - \mathcal{L} \leq \log(2K - 1) \tag{2.3}$$

From Eq. (2.3), we can see that minimizing InfoNCE loss $\mathcal{L}$ is equivalent to maximizing InfoNCE bound $\hat{I}(h_X; h_Y)$.

Detailed derivation of Eq. 2.3: The inequality is well-known as described in (Oord, Li, and Vinyals 2018). We provide the derivation for a typical augmentation (SimCLR-like augmentation) to make it clear that $\log(2K-1)$, instead of $\log(K)$, is due to the number of terms in the denominator.

K is the batch size and $q(x|y) = \frac{p(x)}{Z(y)} e^{\text{sim}(f(x), f(y))/\tau}$, where $Z(y) = \mathbb{E}_{p(y)}[e^{\text{sim}(f(x), f(y))/\tau}]$; $f = f_p \circ f_e$, where f_e is the encoder network and f_p is the projection head; $\text{sim}(u, v) = u^T v / ||u|| ||v||$ denotes the dot product between l_2 normalized u and v (i.e. cosine similarity); and τ denotes a temperature parameter.

$$\begin{aligned}
I(X;Y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(x|y)}{p(x)}\right] && (2.4)\\
&= \mathbb{E}_{p(x,y)}\left[\log \frac{q(x|y)}{p(x)}\right] + \mathbb{E}_{p(y)}[KL(p(x|y)||q(x|y))] && (2.5)\\
&\geq \mathbb{E}_{p(x,y)}\left[\log \frac{q(x|y)}{p(x)}\right] && (2.6)\\
&= \mathbb{E}_{p(x,y)}\left[\log \frac{e^{\mathrm{sim}(f(x),f(y))/\tau}}{Z(y)}\right] && (2.7)\\
&\approx \mathbb{E}\left[\log \frac{e^{\mathrm{sim}(f(x_i),f(y_i))/\tau}}{\frac{1}{2K-1}\sum_{j=1}^{2K}\mathbb{1}_{[j\neq i]}e^{\mathrm{sim}(f(x_i),f(y_j))/\tau}}\right] && (2.8)\\
&= \log(2K-1) + \mathbb{E}\left[\log \frac{e^{\mathrm{sim}(f(x_i),f(y_i))/\tau}}{\sum_{j=1}^{2K}\mathbb{1}_{[j\neq i]}e^{\mathrm{sim}(f(x_i),f(y_j))/\tau}}\right] && (2.9)\\
&= \log(2K-1) - \mathcal{L} && (2.10)\\
&\triangleq \hat{I}(X;Y) && (2.11)
\end{aligned}$$

The inequality in Eq. (2.6) is due to the non-negativeness of KL-divergence, and the approximation in Eq. (2.8) is due to the replacement of the expectation with its empirical mean. Finally, the negative loss $-\mathcal{L}(x_i)$ in Eq. (2.10) is always negative because the argument of the second log term in Eq. (2.9) is always between zero and one. Therefore, $\hat{I}(X;Y) \leq \log(2K-1)$.

2.1.1. Previous works to understand contrastive learning

Some studies have attempted to understand contrastive learning. (Arora et al. 2019) presented a theoretical framework for analyzing contrastive learning by introducing latent classes and showing provable guarantees on the performance of the learned representations under some conditions. (Purushwalkam and Gupta 2020) aimed to demystify unsupervised contrastive learning by emphasizing the relationship between data augmentation and the corresponding invariances. (Tian et al. 2020) investigated the task-dependent optimal views of contrastive learning from a mutual information perspective. (Wang and Isola 2020) attempted to understand contrastive learning with the use of two important properties: alignment and uniformity. (Wang and Liu 2021) suggested that contrastive loss is a hardness-aware loss function and that temperature controls the strength of penalties on hard negative samples. Our study aims to widen the understanding of contrastive learning by investigating the common beliefs on mutual information.

2.2. Mutual Information

The concept of information is too broad to be captured by a single definition. However, for any probability distribution, we define a quantity called *entropy*. Entropy has many properties that agree with intuitive notions of what the measure of information should be. This concept is extended to define *mutual information*, which measure the amount of information one variable contains about another variable. Then, entropy becomes the self-information of a random variable. As a special case of a more general quantity called *relative entropy* and measures the distance between two probability distributions. For a full review, see (Cover 1999).

Mutual Information (MI) is a well-known metric for estimating the relationship between pairs of variables. MI is a reparameterization-invariant measure, so it can capture even non-linear dependency. MI between two random variables X and Y is defined as follows.

$$I(X;Y) \triangleq KL(p(x,y)||p(x)p(y)) = \mathbb{E}_{p(x,y)} \log \left[\frac{p(x,y)}{p(x)p(y)} \right] \tag{2.12}$$

The definition of MI $I(X;Y)$ can be rewritten as

$$\begin{aligned} I(X;Y) &= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \\ &= \sum_{x,y} p(x,y) \log \frac{p(x|y)}{p(x)} \\ &= -\sum_{x,y} p(x,y) \log p(x) + \sum_{x,y} p(x,y) \log p(x|y) \\ &= -\sum_{x} p(x) \log p(x) - \left(-\sum_{x,y} p(x,y) \log p(x|y) \right) \\ &= H(X) - H(X|Y) \end{aligned}$$

where $H(X)$ is the entropy of X. Thus, $I(X;Y)$ can be interpreted as the reduction in the uncertainty of X due to the knowledge of Y. By symmetry, it also follows that

$$I(X;Y) = H(Y) - H(Y|X).$$

Thus, X says as much about Y as Y says about X.

Further, resulting from $H(X,Y) = H(X) + H(Y|X)$ (Cover 1999), we have

$$I(X;Y) = H(X) + H(Y) - H(X,Y).$$

Finally, self-information (the mutual information of a random variable with itself) is equivalent to the entropy of the random variable.

$$I(X;X) = H(X) - H(X|X) = H(X)$$

We summarize the most frequently used properties of mutual information as follows.

- $I(X;Y) = H(X) - H(X|Y)$
- $I(X;Y) = H(Y) - H(Y|X)$
- $I(X;Y) = H(X) + H(Y) - H(X,Y)$
- $I(X;Y) = I(Y;X)$

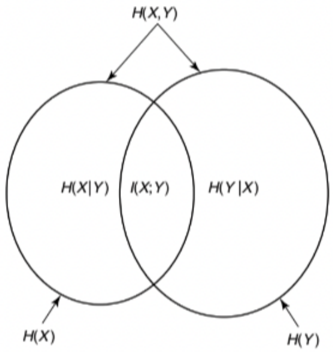


Figure 2.2. Illustration of the relationship between entropy and mutual information. (The figure was adapted from Figure 2.2 of (Cover 1999).)

- $I(X;X) = H(X)$

MI is often referenced as a useful metric or motivation for deep learning. For example, (Xu and Raginsky 2017) derived non-vacuous generalization bounds, (Tishby and Zaslavsky 2015; Achille and Soatto 2018) analyzed the neural network training dynamics, (Moyer et al. 2018) encouraged representation invariance. (Bell and Sejnowski 1995; Hyvärinen and Oja 2000; Oord, Li, and Vinyals 2018; Hjelm et al. 2018) suggested InfoMax principle for improving the representation learning, based on MI.

Although MI has widespread usage, its exact evaluation over high-dimensional variables is almost always impossible because we cannot assess the true probability distribution and only a finite number of samples. There have been some efforts to approximate the probability density function, such as simple binning (Fraser and Swinney 1986; Shwartz-Ziv and Tishby 2017) (discretizing the input variable into a finite number of bins) and non-parametric kernel-density estimators (Kraskov, Stögbauer, and Grassberger 2004). However, they still suffer from the curse of dimensionality and computation cost also largely increases with the data size. To make use of scalable mutual information estimation, variational approaches are suggested as follows.

2.3. Variational Mutual Information Estimators

When only a set of joint samples is available, the exact MI cannot be calculated but an estimation can be made. Among the known MI estimation methods, variational estimators based on variational bounds and DNN modeling have become dominant for the complex datasets. In this paper, we summarize the variational MI estimators following the unified framework of (Poole et al. 2019). For clarity, we follow most of the mathematical descriptions in (Poole et al. 2019).

As an initial lower bound on mutual information, we can simply use the non-negativity of the KL divergence as shown in (Barber and Agakov 2003).

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{q(x|y)}{p(x)}\right] + \mathbb{E}_{p(y)}\left[KL(p(x|y)||q(x|y))\right] \\ &\geq \mathbb{E}_{p(x,y)}\left[\log q(x|y)\right] + h(X) \triangleq I_{\text{BA}}, \end{aligned}$$

where $h(X)$ is the differential entropy of X. The bound is tight when $q(x|y) = p(x|y)$. Unfortunately, we cannot assess $h(X)$ and $q(x|y)$ is also intractable, in general.

To derive tractable bounds that do not use $h(X)$ and $q(x|y)$, we can use unnormalized distributions for the variational family of $q(x|y)$. We consider an energy-based variational family that uses a *critic* $f(x,y)$ and is scaled by the data density $p(x)$:

$$q(x|y) = \frac{p(x)}{Z(y)} e^{f(x,y)},$$

where $Z(y) = \mathbb{E}_{p(x)}\left[e^{f(x,y)}\right]$.

Then, we can establish the unnormalized version of I_{BA} as

$$\mathbb{E}_{p(x,y)}\left[f(x,y)\right] - \mathbb{E}_{p(y)}\left[\log Z(y)\right] \triangleq I_{\text{UBA}}.$$

This bound is tight when $f(x,y) = \log p(y|x) + c(y)$, and $c(y)$ is dependent on y only. However, the log partition function, $\log Z(y)$, is still intractable. (Donsker and Varadhan 1983) applied Jensen's inequality to $\mathbb{E}_{p(y)}\left[\log Z(y)\right]$ and obtained I_{DV} as:

$$I_{\text{UBA}} \geq \mathbb{E}_{p(x,y)}[f(x,y)] - \log \mathbb{E}_{p(y)}[Z(y)] \triangleq I_{\text{DV}}.$$

When we set the critic $f_\theta(x, y)$ as the deep network with parameters θ and train $f_\theta(x, y)$ based on stochastic gradient descent with a mini-batch size of K, the gradient becomes

$$\hat{G}_K = \mathbb{E}_K[\nabla_\theta f_\theta] - \frac{\mathbb{E}_K\left[\nabla_\theta f_\theta e^{f_\theta}\right]}{\mathbb{E}_K\left[e^{f_\theta}\right]}.$$

In the second term of $\hat{G}_K$, the expectations over the samples of a mini-batch K lead to a biased estimation of the full batch gradient. To reduce the bias, (Belghazi et al. 2018) (I_{MINE}) replaced the estimate in the denominator using exponential moving average as below.

$$\hat{G}'_K = \mathbb{E}_K[\nabla_\theta f_\theta] - \frac{\mathbb{E}_K\left[\nabla_\theta f_\theta e^{f_\theta}\right]}{\mathbb{E}_K\left[e^{f_\theta}\right]} * \frac{\mathbb{E}_K\left[e^{f_\theta}\right]}{EMA\left[e^{f_\theta}\right]}.$$

I_{MINE} revises the gradients during training the critic network, while I_{DV} is used for estimation.

To derive a tractable lower bound, we can bound the log partition function, $\log Z(y)$, using the inequality $\log{(x)} \leq \frac{x}{a} + \log{(a)} - 1$ for all $x, a > 0$. When we apply $\log Z(y)$ in I_{UBA}, we obtain a Tractable Unnormalized version of Barber and Agakov (TUBA) lower bound on MI (Poole et al. 2019).

$$I_{\text{UBA}} \geq \mathbb{E}_{p(x,y)}\left[f(x, y)\right] - \mathbb{E}_{p(y)}\left[\frac{\mathbb{E}_{p(x)}\left[e^{f(x,y)}\right]}{a(y)} + \log{(a(y))} - 1\right] \triangleq I_{\text{TUBA}}$$

This bound becomes tighter when we maximize I_{TUBA} with respect to $a(y)$ and f.

I_{TUBA} holds for any $a(y) > 0$, and we can simplify the existed MI estimators by adopting different $a(y)$. When we set $a(y) = e$ (constant), I_{TUBA} recovers the bound of I_{NWJ} (Nguyen, Wainwright, and Jordan 2010), also known as f-GAN KL (Nowozin, Cseke, and Tomioka 2016) and MINE-f (Belghazi et al. 2018).

$$\mathbb{E}_{p(x,y)}[f(x, y)] - e^{-1}\mathbb{E}_{p(y)}[Z(y)] \triangleq I_{\text{NWJ}}$$

I_{NWJ} yields a unique optimal critic $f^*(x, y) = 1 + \log \frac{p(x|y)}{p(x)}$.

When we set $a(y) = \frac{1}{K}\sum_{i=1}^{K} e^{f(x_i,y)}$ to be dependent on multiple samples of x_i, I_{TUBA} recovers the bound of I_{infoNCE} (Oord, Li, and Vinyals 2018).

$$I(X;Y) \geq \mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K} \log \frac{e^{f(x_i,y_i)}}{\frac{1}{K}\sum_{j=1}^{K} e^{f(x_i,y_j)}}\right] \triangleq I_{\text{infoNCE}}$$

where the expectation is over K independent samples from the joint distribution, $\Pi_j p(x_j, y_j)$. The optimal critic for I_{infoNCE} is $f^*(x,y) = \log p(y|x) + c(y)$, where $c(y)$ is any function that depends on y but not x (Ma and Collins 2018). I_{infoNCE} is known as a high-bias low-variance bound because it is upper bounded by $\log K$. Thus, when we use I_{infoNCE} to estimate large MI, we require a large batch size.

For a variational MI estimator, a DNN is used for modeling the critic function $f(x,y)$, and there are two associated steps. The first is the optimization (or training) step where the DNN parameters are learned. The second is the estimation step where the actual MI values are inferred with the optimized DNN. Variational MI estimators such as DV, NWJ, and infoNCE use a single loss function for both optimization and estimation, and the loss function corresponds to the theoretical MI bound in use. Other variational MI estimators, such as JS, MINE, and SMILE, adopt small modifications in either optimization or estimation to improve the robustness or accuracy of the estimator. The most popular variational MI estimators are summarized in Table 2.1. We also can define the variational bounds of mutual information as follows.

Definition 1 (Variational bounds of MI (Poole et al. 2019))**.** *Let X, Y be two random variables taking values in $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N} \sim X, Y$ denotes the set of samples drawn from a joint distribution over $\mathcal{X}$ and $\mathcal{Y}$. The variational bounds of $I(X;Y)$ are formulated as:*

$$\begin{aligned} I(X;Y) &\geq \hat{I}(X;Y) \\ &= 1 + \mathbb{E}_{p(x,y)}\left[\log \frac{e^{f(x,y)}}{a(y)}\right] - \mathbb{E}_{p(x)p(y)}\left[\frac{e^{f(x,y)}}{a(y)}\right] \end{aligned}$$

where $a(y) > 0$ is any value or function of y. The MI estimators are defined by adopting different $a(y)$. For example, $a(y) = e$ (constant) for $\hat{I}_{NWJ}(X;Y)$ (Nguyen, Wainwright, *and Jordan 2010) (also known as f-GAN KL (Nowozin, Cseke, and Tomioka 2016) and MINE-f (Belghazi et al. 2018)) and $a(y) = \frac{1}{K}\sum_{i=1}^{K} e^{f(x_i,y)}$ for $\hat{I}_{\textit{infoNCE}}(X;Y)$ (Oord, Li, and Vinyals 2018). (For a full review, see (Poole et al. 2019).)*

Table 2.1. Summary of variational mutual information estimators. For the optimization step, we find $f^*(x,y)$ to maximize the optimization loss $\mathcal{L}(f(x,y))$ for a given batch size K. For the estimation step, we evaluate the MI values with the variational bounds.

Estimator	Optimization Loss - $\mathcal{L}(f(x,y))$	Estimate Evaluation - $\hat{I}(X;Y)$
DV	$\mathcal{L}_{DV}(f(x,y)) = \hat{I}_{DV}(X;Y) = \mathbb{E}_{p(x,y)}[f(x,y)] - \log \mathbb{E}_{p(x)p(y)}[e^{f(x,y)}]$	(Donsker and Varadhan 1983)
NWJ	$\mathcal{L}_{NWJ}(f(x,y)) = \hat{I}_{NWJ}(X;Y) = \mathbb{E}_{p(x,y)}[f(x,y)] - e^{-1}\mathbb{E}_{p(x)p(y)}[e^{f(x,y)}]$	(Nguyen, Wainwright, and Jordan 2010)
infoNCE	$\mathcal{L}_{infoNCE}(f(x,y)) = \hat{I}_{infoNCE}(X;Y) = \mathbb{E}_{p^B(x,y)}\left[\frac{1}{B}\Sigma_{i=1}^{B} \log \frac{f(x_i,y_i)}{\frac{1}{B}\Sigma_{j=1}^{B} f(x_i,y_j)}\right]$	(Chen et al. 2018)
JS (Poole et al. 2019)	$\mathbb{E}_{p(x,y)}\left[-\text{Softplus}(-f(x,y))\right] - \mathbb{E}_{p(x)p(y)}\left[\text{Softplus}(f(x,y)\right]$	$\hat{I}_{NWJ}(X;Y)$
MINE (Belghazi et al. 2018)	$\mathbb{E}_{p(x,y)}[f(x,y)] - \frac{\mathbb{E}_{p(x)p(y)}[e^{f(x,y)}]}{\text{ExponentialMovingAverage}(\mathbb{E}_{p(x)p(y)}[e^{f(x,y)}])}$	$\hat{I}_{DV}(X;Y)$
SMILE (Song and Ermon 2019)	$\mathcal{L}_{JS}(f(x,y))$	$\mathbb{E}_{p(x,y)}[f(x,y)] - \log \mathbb{E}_{p(x)p(y)}[\text{clip}(e^{f(x,y)}, e^{-\tau}, e^{\tau})]$

2.3.1. Critic function

For a variational MI estimator, a DNN is used to model the *critic* function $f(x,y)$, and there are two associated steps. The first is the optimization (or training) step where the DNN parameters are learned. The second is the estimation step where the actual MI values are inferred with the optimized DNN. Variational MI estimators, such as DV (Donsker and Varadhan 1983), NWJ (Nguyen, Wainwright, and Jordan 2010), and infoNCE (Oord, Li, and Vinyals 2018), use a single loss function for both optimization and estimation, and the loss function corresponds to the theoretical MI bound in use. Other variational MI estimators, such as JS (Nowozin, Cseke, and Tomioka 2016), MINE (Belghazi et al. 2018), and SMILE (Song and Ermon 2019), adopt small modifications in either optimization or estimation to improve the robustness or accuracy of the estimator. The most popular variational MI estimators are summarized in Table 2.1.

Common choices for the critic function $f(x, y)$ include (1) the inner product critic $f_{\text{inner}}(x_i, y_j) = x_i^T y_j$, (2) bilinear critic $f_{\text{bi}}(x_i, y_j) = x_i^T W y_j$ where W is trainable, (3) separable critic $f_{\text{sep}}(x_i, y_j) = f_1(x_i)^T f_2(y_j)$, and (4) joint critic $f_{\text{joint}}(x_i, y_j) = f([x_i, y_j])$. Here f_1, f_2, f are typically shallow MLPs. Critic functions calculate the relationship between all pairs of $(x_i, y_j)\ \forall i, j \in [1, K]$, and the result is a matrix as given below.

$$f(x, y) = \begin{pmatrix} f(x_1, y_1) & f(x_1, y_2) & \cdots & f(x_1, y_K) \\ f(x_2, y_1) & f(x_2, y_2) & \cdots & f(x_2, y_K) \\ \vdots & \vdots & \ddots & \vdots \\ f(x_K, y_1) & f(x_K, y_2) & \cdots & f(x_K, y_K) \end{pmatrix}$$

Variational bounds approximate MI by using the diagonal terms as the values from the joint distribution $p(x, y)$ and the off-diagonal terms as the values from the marginal distribution $p(x)p(y)$.

2.3.2. Limitations of the variational MI estimators

Variational MI estimators present some disadvantages because we typically have access to samples, but not to the underlying distributions (Poole et al. 2019; Song and Ermon 2019; Paninski 2003; McAllester and Stratos 2020). Most estimators exhibit poor performance, particularly when the batch size K is small and the MI is large. For instance, infoNCE results in a high bias because it is upper bounded by $\log K$ (Oord, Li, and Vinyals 2018). (McAllester and Stratos 2020) noted that any distribution-free high-confidence lower bound on MI cannot be larger than $\mathcal{O}(\log K)$. By contrast, most estimators (except infoNCE) result in a variance that can increase exponentially with true MI (Poole et al. 2019; Song and Ermon 2019; Xu et al. 2019). In previous studies, the limitations of estimators have been assessed only for the Gaussian dataset. Although (Song and Ermon 2019) defined a self-consistency test based on the MNIST and CIFAR-10 datasets, they only evaluated the approximated metrics, not the estimation error itself. In this study, we define three factors not considered in the previous studies, which can affect true MI and its estimation. Additionally, we empirically show how variational approaches are accurate for estimating MI based on image and text datasets when their true MI values are accessible.

Chapter 3. Same-class Sampling for Positive Pairing

In contrastive learning, the chosen family of augmentations $\mathcal{T}_{\text{aug}}$ plays the critical role of implicitly determining the joint distribution $p(x, y)$ and marginal distribution $p(x)p(y)$. For the actual training, however, we do not need to know the exact distributions. Instead, we just need to be able to sample with the distributions. Therefore, the concept of augmentation ($\mathcal{T}_{\text{aug}}$) can be expanded to the concept of positive pairing ($\mathcal{T}$), as shown in Figure 3.1. Positive pairing can be performed with an augmentation function, as shown in Figure 3.1(b), or without any augmentation function, as shown in Figure 3.1(c).

In our study, we heavily rely on a simple yet special positive pairing method called *same-class sampling*, $\mathcal{T}_{\text{class}}$. As shown in Figure 3.1(c), same-class sampling only relies on the downstream task's label information and does not utilize any augmentation. Same-class sampling is special because the only shared information between the two

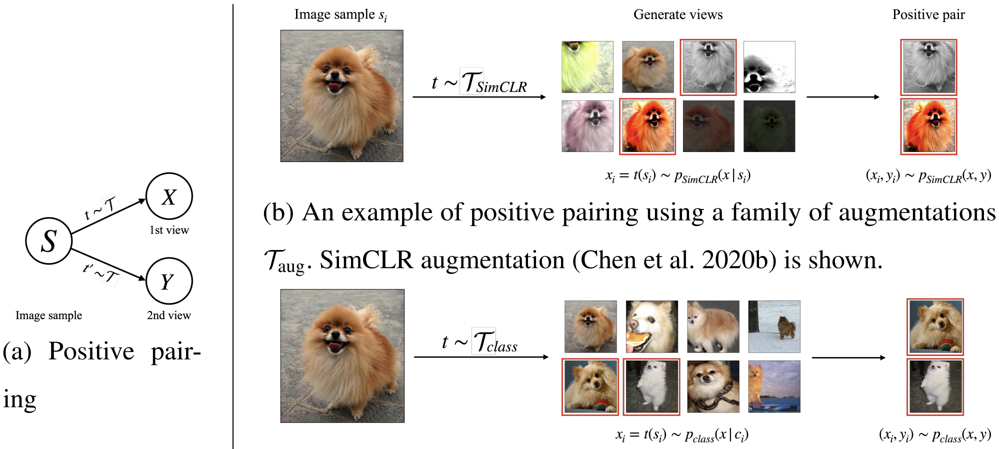


Image sample s_i
Generate views
Positive pair
$t \sim \mathcal{T}_{SimCLR}$
$x_i = t(s_i) \sim p_{SimCLR}(x\,|\,s_i)$
$(x_i, y_i) \sim p_{SimCLR}(x, y)$
S
X
Y
$t \sim \mathcal{T}$
$t' \sim \mathcal{T}$
1st view
2nd view
Image sample
$t \sim \mathcal{T}_{class}$
$x_i = t(s_i) \sim p_{class}(x\,|\,c_i)$
$(x_i, y_i) \sim p_{class}(x, y)$

(a) Positive pairing

(b) An example of positive pairing using a family of augmentations $\mathcal{T}_{\text{aug}}$. SimCLR augmentation (Chen et al. 2020b) is shown.

(c) Positive pairing with same-class sampling $\mathcal{T}_{\text{class}}$. Unlike the case of using a family of augmentations $\mathcal{T}_{\text{aug}}$, only the downstream task's class information is used for positive pairing.

Figure 3.1. Positive pairing method implicitly determines the joint and marginal distributions – $p(x, y)$ and $p(x)p(y)$ are determined by the choice of $\mathcal{T}$.

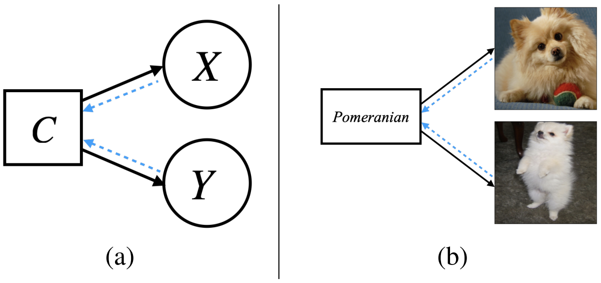


Figure 3.2. Illustration of same-class sampling. (a) The black solid arrows refer to same-class sampling. We sample the positive pair (x_i, y_i) to share the information source c_i and no additional information. The blue dotted arrows refer to an assumption of Theorem 2 whereby an error-free classification function $h_{\text{class}} : X \rightarrow C$ exists. (b) An example of ImageNet.

views is the downstream task's class information. In this case, the true MI for its joint distribution $p_{\text{class}}(x, y)$ can be proven to be upper bounded by the entropy of the class distribution, $H(C)$. We provide the proof below.

Theorem 1 (Same-class sampling for positive pairing)**.** *When we generate inputs X and Y based on the information source C following the Markov process $X \leftarrow C \rightarrow Y$, then $I(X;Y) \leq H(C)$.*

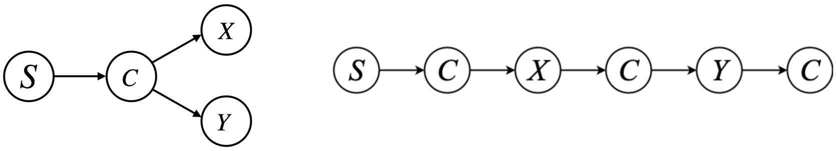
Proof. From the construction of same-class sampling, $X \leftarrow C \rightarrow Y$, the dependency is Markov equivalent to $X \rightarrow C \rightarrow Y$. Then,

$$I(X;Y) \leq I(X;C) = H(C) - H(C|X) \leq H(C)$$

where the first inequality follows from the data processing inequality and the second inequality follows from the positiveness of entropy for the case of discrete random variable C. □

We also provide a stronger result of an equality proof under a mild assumption. Note that we could omit the anchor S when we generalize our statement to setups other than contrastive learning.

Theorem 2 (Same-class sampling for positive pairing (Equality))**.** *When we generate inputs X and Y based on the information source C following the Markov process $X \leftarrow C \rightarrow Y$, and there exists a perfect classification function f_{class} that outputs C from X and Y (i.e. $X \rightarrow C$ and $Y \rightarrow C$), then $I(X;Y) = H(C)$.*



(a)

(b)

Figure 3.3. Markov process of same-class-sampling, $\mathcal{T}_{\text{class}}$. S denotes the anchor image, C denotes the image's downstream task class label, and X and Y correspond to the positive pair chosen for same-class-sampling of the image S. (a) The original Markov process of same-class-sampling. (b) Equivalent Markov process of the same-class-sampling.

Proof. The Markov dependency of $\mathcal{T}_{\text{class}}$ can be summarized, as shown in Figure 3.3(a). For the same-class-sampling, C is the common class label of S, X, and Y.

Assumption: $c_i = f_{class}(s_i) = f_{class}(x_i) = f_{class}(y_i)$, where $f_{class}(\cdot)$ is a function that returns the class label information.

When we have an accurate classifier $f_{class}(.)$ as described in the assumption, $H(C|X) = 0$ and $H(C|Y) = 0$. Due to the deterministic nature of each image's class label.

$$I(X;C) = H(C) - H(C|X) = H(C) \tag{3.1}$$

$$I(Y;C) = H(C) - H(C|Y) = H(C) \tag{3.2}$$

Because C can be perfectly determined from either X or Y, the Markov process in Figure 3.3(a) can be alternatively expressed as $S \rightarrow C \rightarrow X \rightarrow C \rightarrow Y \rightarrow C$, as shown in Figure 3.3(b). Here, the first part of the new Markov process is the same as in Figure 3.3(a): we start from s_i, read its class label $c_i = f_{class}(s_i)$, and sample an example x_i using the class label c_i. In Figure 3.3(b), however, we can alternatively read x_i's class label without any uncertainty to recover $c_i = f_{class}(x_i) = f_{class}(s_i)$ and then use the class label to sample y_i. Because $c_i = f_{class}(y_i)$ can be recovered from y_i, the last part of dependency, $Y \to C$, follows. For the equivalent Markov process in Figure 3.3(b), we derive an upper bound and a lower bound to complete the proof.

Upper bound: We apply the data processing inequality (Cover 1999) to the Markov dependency $X \to C \to Y$ in the middle part of Figure 3.3(b).

$$I(X;Y) \leq I(X;C) \tag{3.3}$$

$$= H(C) - H(C|X) \tag{3.4}$$

$$= H(C) \tag{3.5}$$

Eq. (3.3) is the data processing inequality, Eq. (3.4) is from the definition of MI, and Eq. (3.5) is because of $H(C|X) = 0$ as in Eq. (3.1).

Lower bound: We apply the data processing inequality (Cover 1999) to the Markov dependency $C \to X \to C \to Y \to C$ part of Figure 3.3(b). The following directly follows from the data processing inequality.

$$I(C;C) \leq I(X;Y) \tag{3.6}$$

$$\Rightarrow H(C) \leq I(X;Y) \tag{3.7}$$

Note that we have C in the beginning and at the end of the Markov dependency. The first C in $I(C;C)$ corresponds to the C in the beginning, and the second C in $I(C;C)$ corresponds to the C at the end of the Markov dependency. Eq. (3.7) is because $I(C;C)$ is the self-information that is the same as $H(C)$.

Therefore, the true mutual information value of same-class-sampling, $I_{\text{class}}(X;Y)$, is the same as the class label's entropy, $H(C)$. □

The calculation of $H(C)$ is trivial for uniformly distributed class labels, and the result indicates that the class information is the only meaningfully shared information between a pair of positive examples. We denote the true MI as $I_{\text{class}}(h_X; h_Y)$ and its estimate as $\hat{I}_{\text{class}}(h_X; h_Y)$.

The $H(C)$ upper bound on same-class sampling reveals that the downstream-task information, with its entropy $H(C)$, is the only meaningfully shared information between a pair of positive examples. This result can be conveniently utilized in our empirical investigations because the calculation of $H(C)$ is trivial for uniformly distributed class labels. Note that same-class sampling is a supervised method because it utilizes class information. *We are introducing this supervised method only for the purpose of theoretical study and empirical investigation, and we are not suggesting its use for a practical purpose.* We denote the true MI of same-class sampling as $I_{\text{class}}(h_X; h_Y)$ and its estimate as $\hat{I}_{\text{class}}(h_X; h_Y)$.

By introducing the same-class sampling, we can (1) examine how the estimations are accurate based on the true MI value, and (2) make the views share only the downstream-task relevant information.

Unlike the same-class sampling, MI of augmentation-based methods such as $\mathcal{T}_{\text{SimCLR}}$ (Chen et al. 2020b), $\mathcal{T}_{\text{AutoAugment}}$ (Cubuk et al. 2018) and $\mathcal{T}_{\text{RandAugment}}$ (Cubuk et al. 2020) are intractable because the shared information is dependent on the particular choice of $\mathcal{T}_{\text{aug}}$ whose joint distribution is unknown. Furthermore, the shared information does not need to be relevant with the downstream-task performance. Let us articulate on the example in Figure 3.4. When we use the general augmentation, the true MI is not available because of the intractable joint distribution $p(x, y)$, and the shared information seems somewhat irrelevant to the label information. On the contrary, when we use same-class sampling, the true MI is upper bounded as the entropy of the label (or equivalent to the entropy of the label), and the shared information should be relevant to the label information. In this study, we select $\mathcal{T}_{\text{SimCLR}}$ as the representative example of $\mathcal{T}_{\text{aug}}$ because it has been widely used in previous works (Chen et al. 2020c; Chen and He 2021; Caron et al. 2020; Grill et al. 2020b; Zbontar et al. 2021; Bardes, Ponce, and LeCun 2021; Tomasev et al. 2022).

Remark: Same-class sampling requires the discretized variable C. Even though the same-class sampling provides the (upper bound of or the exact) true MI values for any type of dataset, the discretized variable C is required. The use of C has a clear trade-off. We first define the joint distribution $p(x, y)$ depending on the specific downstream task. In a later section, we utilize this property to examine some existing beliefs to be corrected. Conversely, different choices of C result in different MI values. Thus, our results should not be considered as the general property of MI, and it can be largely dependent on the particular choice of positive pairing method. Thus, we note that our analysis framework can be a guideline for utilizing mutual information in understanding representation learning, but it also has some limitations.

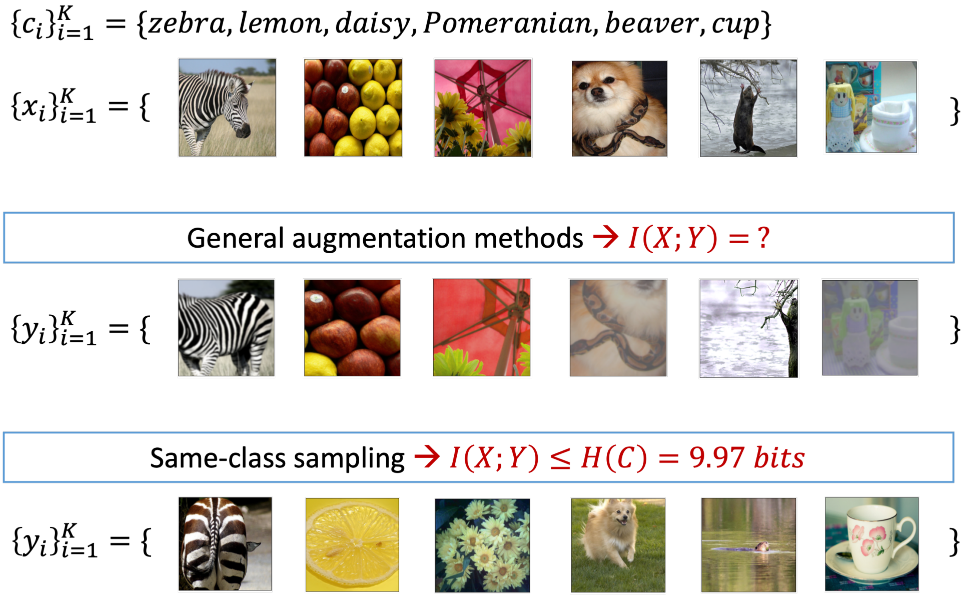

$\{c_i\}_{i=1}^{K} = \{zebra, lemon, daisy, Pomeranian, beaver, cup\}$
$\{x_i\}_{i=1}^{K} = \{$ $\}$
General augmentation methods → $I(X;Y) = ?$
$\{y_i\}_{i=1}^{K} = \{$ $\}$
Same-class sampling → $I(X;Y) \leq H(C) = 9.97$ $bits$
$\{y_i\}_{i=1}^{K} = \{$ $\}$

Figure 3.4. While we have no access to the true MI value in the case of the general augmentation methods, we can make use of true MI for any type of dataset in the case of same-class sampling. Even though the assumption is not guaranteed, we have access to the upper bound of true MI for any dataset.

Chapter 4. Understanding the Accuracy of Variational Mutual Information Estimators

In representation learning, Mutual Information (MI) has been a popular ingredient for estimating the relationship between pairs of random variables. In particular, optimizing the MI estimator as the loss function of unsupervised learning has improved the downstream task performance across various application domains. While MI is an elegant concept with extensive use cases, its exact evaluation over real-world datasets is almost always impossible because of the unknown $p(x, y)$. Even when the joint probability distribution is known, the evaluation is known as notoriously difficult when the two variables are in high-dimensional space.

To work around these problems, variational MI estimators that can be efficiently combined with deep learning methods have been developed (Belghazi et al. 2018; Poole et al. 2019; Song and Ermon 2019; 2020; Cheng et al. 2020). Variational estimators are based on two steps. First, an analytical bound is derived where the bound is based on a critic function $f(x, y)$. Second, the bound is made tight by optimizing for a supremum or infimum over $f(x, y)$. In recent works, DNNs have been used to model the critic function. When a proper loss function is chosen and the learning of $f(x, y)$ is successful, the variational estimations have been shown to be accurate for a toy dataset.

Even though variational approaches have achieved an improvement in estimation accuracy, they have only considered the Gaussian dataset for evaluation (Belghazi et al. 2018; Poole et al. 2019; Song and Ermon 2019; 2020; Cheng et al. 2020). The toy dataset is convenient because it offers a simple analytical formula for the true MI, $I(X; Y)$. With the true MI, it becomes possible to assess the characteristics of the estimated MI, $\hat{I}(X; Y)$. For instance, accuracy, bias, and variance of variational MI estimators have been analyzed by comparing $I(X; Y)$ and $\hat{I}(X; Y)$ (Belghazi et al. 2018; Poole et al. 2019; Song and Ermon 2019; 2020; Cheng et al. 2020). The Gaussian dataset, however, is far from being representative of real-world datasets, because its underlying structure is purely statistical. For real-world datasets with strong manifold structures, it has been pointed out that variational MI estimators might be inaccurate and misleading (Song and Ermon 2019). It would be ideal if real-world datasets with known true MI were available. In the absence of such datasets, (Song and Ermon 2019) proposed three types of self-consistency tests and found that none of the existing MI estimators passed all the tests. The result, however, is limited in that a direct assessment of estimator accuracy is missing. To address this issue, we adopt and modify a method for generating image and text datasets with known true MI. In fact, we can easily manipulate the true MI. The idea is to pair two inputs according to their labels, such that the true MI is determined as the result of positive pairing.

In this chapter, we examine the accuracy of variational MI estimators under various scenarios. While the previous studies have taken into account only a few factors affecting MI estimation, such as the batch size and true MI, we additionally define three factors, namely the number of information sources, dimension of representations, and nuisance strength. Then, we examine how the individual factors affect the estimation accuracy of variational approaches for a variety of data domains. In this study, we consider the three types of data domains, including the multivariate Gaussian, the images as an example of vision tasks, and the sentence embeddings as an example of NLP tasks. We summarize the main takeaways of this chapter as follows:

1. We define three factors that can affect the true and estimated MI values.

2. We use same-class sampling to access the true MI for any dataset. In fact, we can easily manipulate the true MI values. In this study, we consider the images as an example of vision tasks and the sentence embeddings as an example of NLP tasks.

3. We evaluate the accuracy of six MI estimators for various data domains, including image and text. Because our evaluation is based on true MI, we can analyze the

exact estimation error, instead of the approximation.

4. We summarize the empirical findings for individual factors. We found that several results are not aligned with the previous beliefs, such as increasing the model capacity for estimation does not always improve estimation accuracy.

5. We show how our method is applicable to a practical dataset based on the ImageNet dataset. We can estimate MI with true values when we have a well-trained encoder network that can predict the information sources from the given images.

In conclusion, MI can be an excellent metric for understanding deep representations, but some precautions are required to measure MI based on variational approaches combined with deep networks. We expect our method and empirical results to become the cornerstone of using mutual information for an in-depth understanding of representation learning.

4.1. Datasets

In this study, we evaluate variational MI estimators across various data domains: (1) multivariate Gaussian ($\mathcal{D}_{Gaussian}$), corresponding to the most common case for evaluating MI estimation accuracy (Poole et al. 2019; Song and Ermon 2019; McAllester and Stratos 2020); (2) an image dataset consisting of digits ($\mathcal{D}_{vision}$), as an example of the vision tasks; and (3) sentence embeddings consisting of the BERT embeddings of movie review datasets ($\mathcal{D}_{NLP}$), as an example of NLP tasks. We first introduce the general formulation of a Gaussian dataset for MI estimation. Then, we define three factors that can affect true MI and its estimation, which have not been considered previously. Finally, we develop same-class sampling to make use of true MI without the limitation that the probability distribution should be tractable.

4.1.1. Gaussian dataset

Consider an observational dataset with K pairs of samples, where each pair (x_i, y_i) is sampled from a joint distribution $p(x, y)$. A variational MI estimator utilizes the dataset as its input, and evaluates the estimated mutual information $\hat{I}(X;Y)$. If the estimation is accurate, $\hat{I}(X;Y)$ should be close to the true mutual information $I(X;Y)$. In previous studies, a Gaussian dataset associated with a multivariate Gaussian model was utilized to assess the variational MI estimators (Belghazi et al. 2018; Poole et al. 2019; Song and Ermon 2019; 2020; Cheng et al. 2020). The Gaussian dataset has Gaussian samples with zero mean and component-wise correlation of ρ between X and Y. The true MI is known and it can be expressed analytically as $I(X;Y) = -\frac{d_g}{2}\log(1-\rho^2)$, where $x \in \mathbb{R}^{d_g}$ and $y \in \mathbb{R}^{d_g}$.

4.1.2. Definitions of d_s, d_r, and Z

Among the numerous factors that can affect the mutual information $I(X;Y)$, we focus on the number of information sources, dimension of representation, and nuisance factor. For random variables X and Y with a joint distribution $p(x, y)$, they can be defined as follows.

Definition 2 (Number of information sources, d_s)**.** *d_s is the number of independent scalar random variables used to form the mutually shared information between X and Y.*

Definition 3 (Dimension of representation, d_r)**.** *d_r is the size of the observational data. When X and Y are of the same size, it is the length of the vector formed by flattening either X or Y.*

Definition 4 (Nuisance, Z)**.** *Nuisance to a random variable X is defined as an equal-size random variable Z sharing no information with X. Mathematically, Z satisfies $I(X;Z) = 0$. Nuisance to (X, Y) can be defined similarly where Z is of the same size as (X, Y) and $I(X, Y; Z) = 0$.*

Note that the definition of nuisance is similar to the definition in (Achille and Soatto 2018). Definition 4, however, is simpler because the concept of a task is not involved.

As an example of the above definitions, consider a Gaussian dataset; its number of information sources d_s is equal to d_g, its dimension of representation d_r is also equal to d_g, and the dataset contains no nuisance.

4.1.3. Details of generating datasets

Following Theorem 2, it is possible to assess the true MI for any dataset when we draw the positive samples from the joint distribution $p(x, y)$ that is dependent on the information sources C only. Because there are too many options for generating a dataset, we add the restrictions that d_s, d_r, and η are fixed and only the MI value changes. We first consider the binary random variable C with $p(0) = p(1) = 0.5$. We can design a simple stochastic function that maps C to X, where X is an image or sentence embedding. To make use of the error-free classification function $h_{\text{class}}(\cdot)$ in Theorem 2, we choose a dataset that easily achieves perfect classification accuracy with a simple classifier (e.g., 1-layer MLP). We adopt the MNIST dataset (Deng 2012) for $\mathcal{D}_{vision}$ and BERT (Devlin et al. 2018) fine-tuned sentence embeddings of the IMDB dataset (Maas et al. 2011) for $\mathcal{D}_{NLP}$. In our implementation, x becomes a sample from $\mathcal{D}$ of class 0 when $c = 0$, and a sample from $\mathcal{D}$ of class 1 when $c = 1$. We design a mapping function from C to Y where a different image or text is drawn. For this basic construction, it can be shown that $I(X; Y) = H(C) = 1$ bit. We provide an image example in Figure 4.2(a).

To construct a dataset with larger MI, two straightforward approaches can be used. In Figure 4.2(b), we combine four samples of Figure 4.2(a) to create an image that is four times larger, which means $I(X; Y) = 4$. In Figure 4.2(c), we stack three pairs of samples from Figure 4.2(a) and map them to RGB; hence, $I(X; Y) = 3$. We can flexibly use stratagems to generate a dataset that has a specific value of true MI. Similarly, we generate the text dataset by concatenating the embedding vectors in 1D.

To construct a dataset with non-integer MI value, binary symmetric channel (BSC) is utilized (Cover 1999). Consider the case of Figure 4.2(a). If C is passed through a BSC with crossover probability β, it can be shown that $I_{\text{BSC}}(X;Y) = 1 - H(\beta)$. Then, the MI value can be controlled by adjusting β between 0 and 0.5. Extension to other datasets is trivial, and in general, $I_{\text{BSC}}(X;Y) = I(X;Y) \times (1 - H(\beta))$.

Detailed description of binary symmetric channel (BSC): To construct a dataset with a non-integer MI value, Binary Symmetric Channel (BSC) is utilized (Cover 1999). Consider the case of the images where $I(X;Y) = 1$ (Figure 4.2(a)). As shown in the figure below, the transmission process of BSC for $C \to Y$ corresponds to a binary channel where the input is complemented with probability β. $H(Y|X) = H(Y|C)$ because $H(X|C) = 0$. Then, the mutual information can be evaluated as follows.

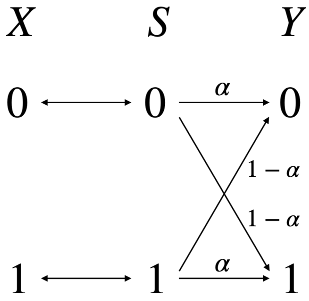


Figure 4.1. Construction of image dataset with non-integer MI value by utilizing the binary symmetric channel.

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - \sum p(x) H(Y|X=x) \\ &= H(Y) - \sum p(x) H(\beta) \\ &= H(Y) - H(\beta) \\ &= H(Y) + \beta \log \beta + (1-\beta) \log (1-\beta). \end{aligned}$$

$H(\beta)$ is symmetric because $H(\beta) = -\beta \log \beta - (1-\beta)\log(1-\beta) = H(1-\beta)$. Therefore, we consider β only for $[0, 0.5]$, instead of $[0, 1]$. Extension to other datasets is trivial and $I_{BSC}(X;Y) = I(X;Y) \times (1 - H(\beta))$ in general.

For images, we can insert random samples from other datasets as nuisance to X and Y to make the dataset more realistic without affecting the true MI value, as shown in Figure 4.2(d). Because the source images remain on top without any occlusion, and there is no fixed relationship between the background chosen for X and the background chosen for Y, the nuisance Z does not affect the true $I(X;Y)$.

For $\mathcal{D}_{vision}$ and $\mathcal{D}_{NLP}$, it is trivial to identify the number of information sources d_s and the dimension of representation d_r. The number of information sources d_s is always equal to $I(X;Y)$. For instance, Figure 4.2(b) has $d_s = 4$. When using BSC, d_s is equal to $I(X;Y)$ that is calculated for $\alpha = 0$. The dimension of representation d_r is a design parameter. As default, we have chosen $d_r = 64^2$ for $\mathcal{D}_{vision}$ and $d_r = 768 \times 10$ for $\mathcal{D}_{NLP}$.

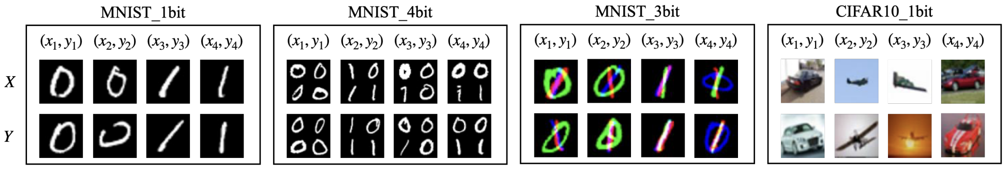


Figure 4.2. Example of generating a dataset with known true MI value for images. X and Y consist of random images drawn from the MNIST dataset. (a) Basic construction: the digits appear together with probability 0.5 and $I(X;Y) = 1$. (b) Concatenate four samples in (x, y)-coordinates: four samples have independent labels and $I(X;Y) = 4$. (c) Concatenate three samples in channel dimension: three samples have independent labels and $I(X;Y) = 3$. (d) Adding nuisance: an independently chosen background image from CIFAR-10 is inserted as the nuisance. Nuisance does not affect the true MI; therefore, $I(X;Y) = 1$.

4.2. Experimental setup

We follow the setup of the case of multivariate Gaussians (Belghazi et al. 2018; Tschannen et al. 2019; Song and Ermon 2019; Poole et al. 2019). In general, the critic network $f(x, y)$ is trained to maximize $\hat{I}(X; Y)$ with the real-time-generated inputs X and Y. When the other factors are fixed, only the true MI ($I(X; Y)$) is controlled during estimation. A complete estimation occurs over 20k steps, and we vary the true MI ($I(X; Y)$) over time.

We provide the architecture details for the critic network as follows. We reference the official code of (Tschannen et al. 2019; Song and Ermon 2019). For the separable critic $f(x_i, y_j) = f_1(x_i)^T f_2(y_j)$, we use the same architecture for f_1 and f_2 as a 2-layer MLP with 256 units and 32-dimensional outputs. For the concatenated critic $f(x_i, y_j) = f([x_i, y_j])$, we use 2-layer MLP with 256 units. To train the critic networks, we set the batch size K as 64. We optimize the variational bounds of mutual information using Adam (Kingma and Ba 2014) with a learning rate of 0.0005.

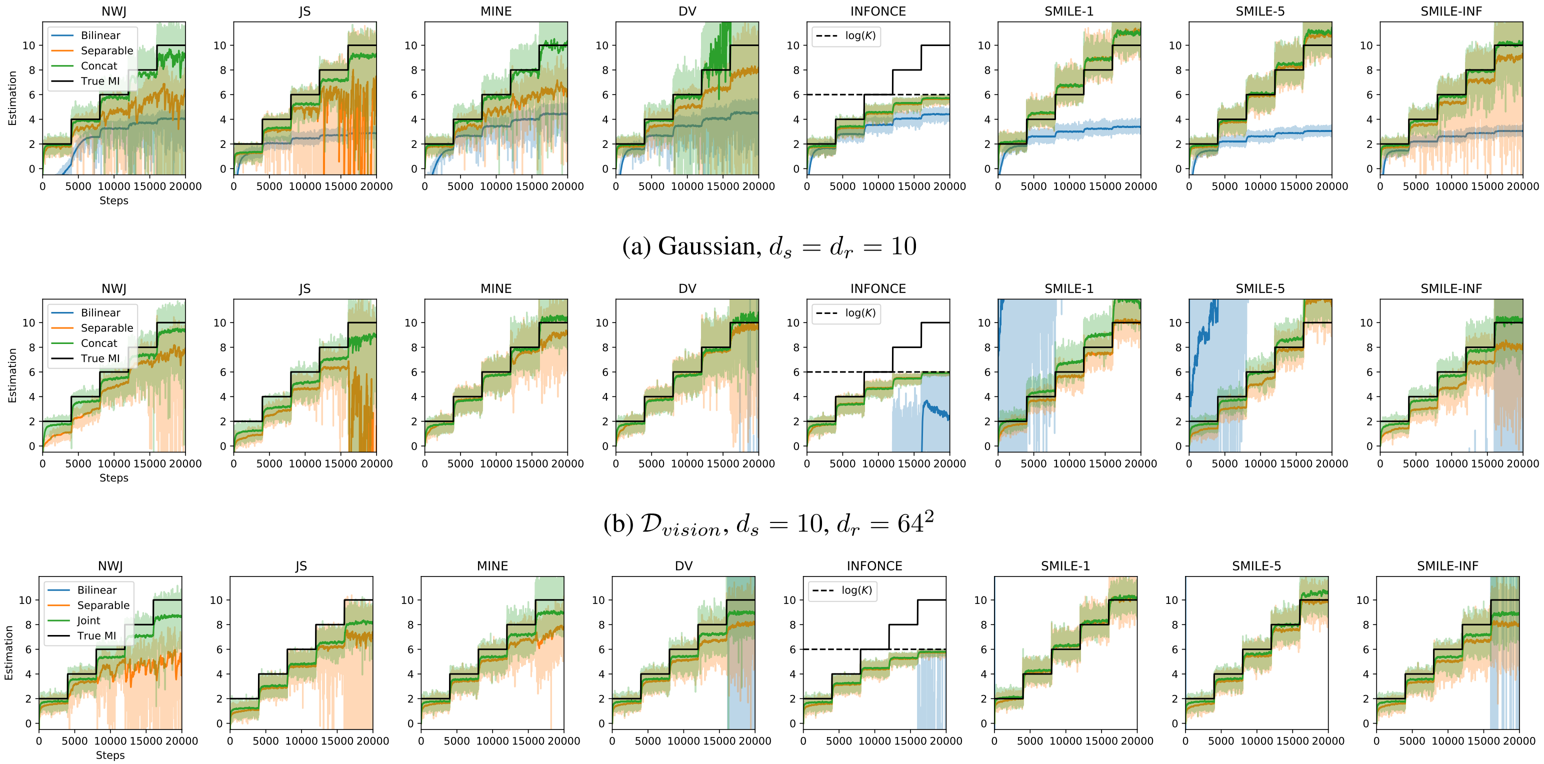
4.3. Experimental results

In this section, we conduct several experiments to examine how the suggested factors affect the estimation accuracy of variational approaches across a variety of data domains.

4.3.1. Critic architecture

We first examine the effect of the choice of critic architecture, which has already been investigated for the Gaussian datasets in (Poole et al. 2019; Song and Ermon 2019). In (Poole et al. 2019), using a joint critic outperforms a separable critic for NWJ and JS estimators, while the InfoNCE estimator is more robust to the choice of critic architecture. For the SMILE estimator (Song and Ermon 2019), using a joint critic outperforms a separable critic for a basic setup of the Gaussian dataset, while the trend is reversed for a more complicated setup of the Gaussian dataset. In this study, we inspect whether the previous findings can be applied to vision and NLP cases.

As shown in Figure 4.3, we found that the joint critic always outperforms the other critics for NWJ and JS estimators for all data domains. However, the improvements become smaller for the NLP case than the Gaussian and the images. For the MINE estimator, using a joint critic clearly outperforms using a separable or bilinear critic for the Gaussian. On the contrary, for vision and NLP cases, the improvements become much smaller, and using a separable critic also provides reasonable results. For the DV estimator, using a joint critic does not result in the best estimation accuracy for the Gaussian dataset. As claimed in (Song and Ermon 2019), the estimation variance can increase exponentially depending on the true MI values. Meanwhile, we found that using a joint critic results in accurate estimations for vision and NLP cases, and the estimation variance does not increase exponentially even for large MI values. We also found that the InfoNCE estimator is quite robust to the choice of critic architecture for all cases because InfoNCE cannot estimate MI larger than $\mathcal{O}(\log K)$. For the SMILE estimator, we found some intriguing results: (1) using a bilinear critic does not result in the lower estimator of MI for vision and NLP cases, while it provides the lower estimator of MI with a high bias for the Gaussian; (2) the SMILE estimator is quite robust to the choice of critic architecture among the joint critic and separable critic for the Gaussian and NLP cases, but using a specific critic (joint critic for SMILE-1 and separable critic for SMILE-5 and SMILE-inf) outperforms the other for the vision case; and (3) the Gaussian and NLP cases are more robust to the choice of tuning parameters for the SMILE estimator (i.e., clipping threshold, τ in Table 2.1) than the vision case.



(a) Gaussian, $d_s = d_r = 10$

(b) $\mathcal{D}_{vision}$, $d_s = 10$, $d_r = 64^2$

(c) $\mathcal{D}_{NLP}$, $d_s = 10$, $d_r = 768 \times 10$

Figure 4.3. Estimation results for three different datasets. We found that the variational estimators of mutual information are surprisingly accurate for image and sentence embeddings, better than Gaussian datasets.

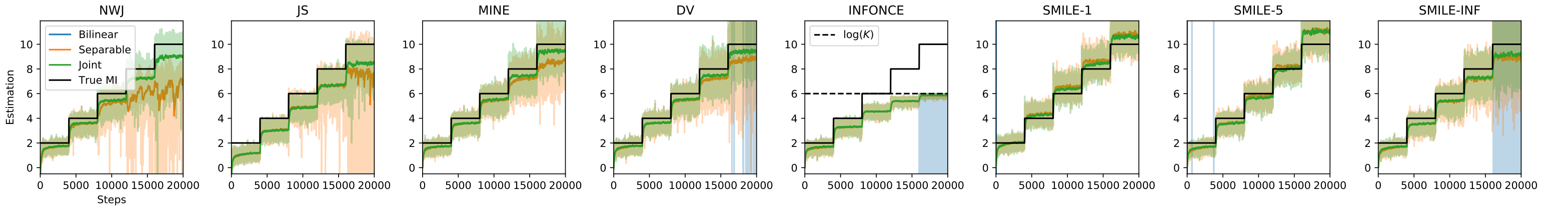
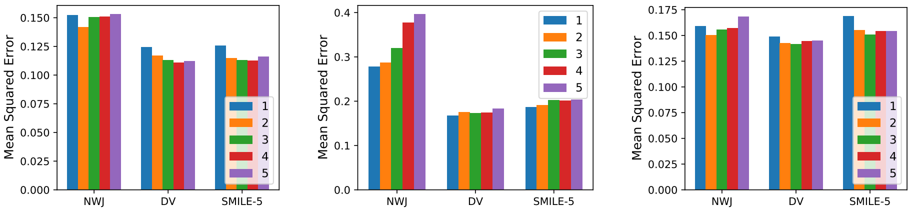


Figure 4.4. Image and sentence embeddings, $d_s = 10$, $d_r = 7680$

In addition to the case of two variables sharing the same data domain of image or text, we test the case of two variables not sharing a domain, i.e., X is an image and Y is text. The results are provided in Figure 4.4, and we observe similar results of the NLP case.

4.3.2. Critic capacity

We observed that using a joint critic generally results in the best estimation accuracy for most of the cases. In this section, we inspect whether increasing the critic capacity of a joint critic could be more beneficial to improve estimation accuracy. In a previous study (Tschannen et al. 2019), a larger critic capacity is known to improve the estimation accuracy. In this study, we increase the critic capacity by increasing the depth of the MLP network, and the results when the true MI is 2 bits are provided in Figure 4.5. Unlike a previous study, we found no positive correlation between critic capacity and estimation accuracy for any data domain. The Pearson's correlation coefficient ρ between the critic capacity and estimation accuracy was $-0.007(\mathcal{D}_{Gaussian})$, $0.059(\mathcal{D}_{vision})$, and $-0.001(\mathcal{D}_{NLP})$. Overall, increasing critic capacity is not always beneficial.



(a) Gaussian (b) Images (c) Sentence embeddings

Figure 4.5. Mean squared error for the different depths of a joint critic function when true MI is 2 bits. For all datasets and estimators, we found no distinguishable improvements when increasing the depth of the critic f_{joint}.

4.3.3. Choice of the variational MI estimator

Recently, there have been some efforts to suggest more accurate variational MI estimators (Poole et al. 2019; Song and Ermon 2019; McAllester and Stratos 2020; Cheng et al. 2020). In particular, the SMILE estimator has been shown to efficiently reduce the estimation variance of other classical estimators (Song and Ermon 2019). Therefore, the SMILE estimator has been accepted as exhibiting better bias-variance trade-offs. As shown in Figure 4.6, we found that the SMILE estimator outperforms the other estimators slightly for the Gaussian case and quite largely for the NLP case. However, we found that the NWJ and MINE estimators outperform the SMILE estimator for large true MI values for the vision case.

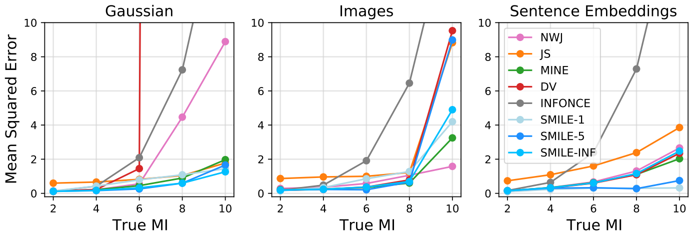


Figure 4.6. Estimation results for three different datasets for different ground-truth MI values. While the SMILE estimator provides a clear improvement for the Gaussian dataset, the MINE and NWJ estimators provide better estimation accuracy for the image dataset.

4.3.4. Number of information sources (d_s)

In this section, we investigate how the number of information sources (d_s), defined in Section 4.1.2, affects the estimation accuracy. For each data domain, we increase d_s from 1 to 100. As shown in Figure 4.7, estimation fails when the number of information sources d_s is large, for all data domains. Evidently, variational estimators start to fail roughly when d_s is increased above 4 for the Gaussian, above 25 for the vision case, and above 36 for the NLP case. The uniformly distributed classification problem

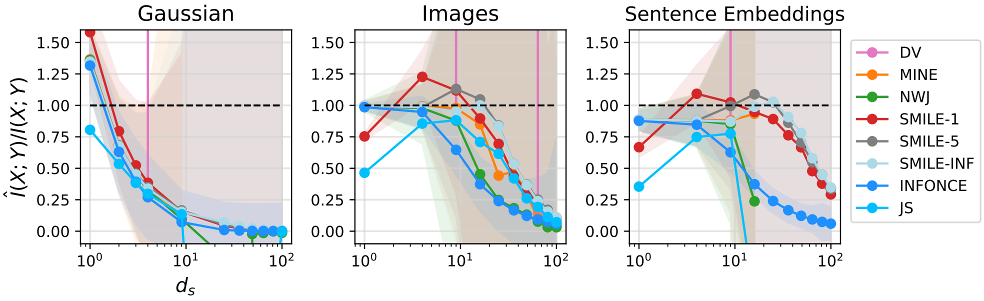


Figure 4.7. Estimation fails when d_s is large. Shades correspond to the standard deviation of the estimations. For all datasets and estimators, the estimation fails when d_s is too large.

corresponds to the number of classes being larger than 10M. Thus, we expect it would not be a limiting factor for practical uses.

4.3.5. Representation dimension (d_r)

We also investigate the influence of representation dimension (d_r), defined in Section 4.1.2. Here, we only consider the vision case, and we increase the representation dimension simply by increasing the output image size from 10^2 to 100^2. As summarized in Figure 4.8, we found that the representation dimension does not affect the estimation accuracy.

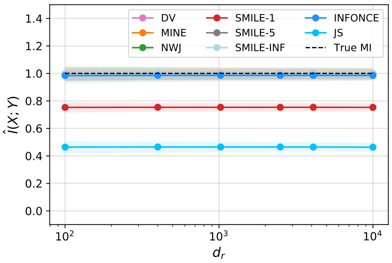


Figure 4.8. MI estimation results when representation dimension d_r is increased.

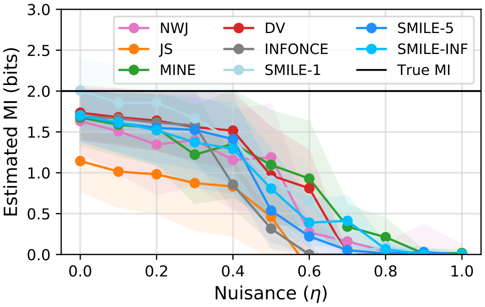


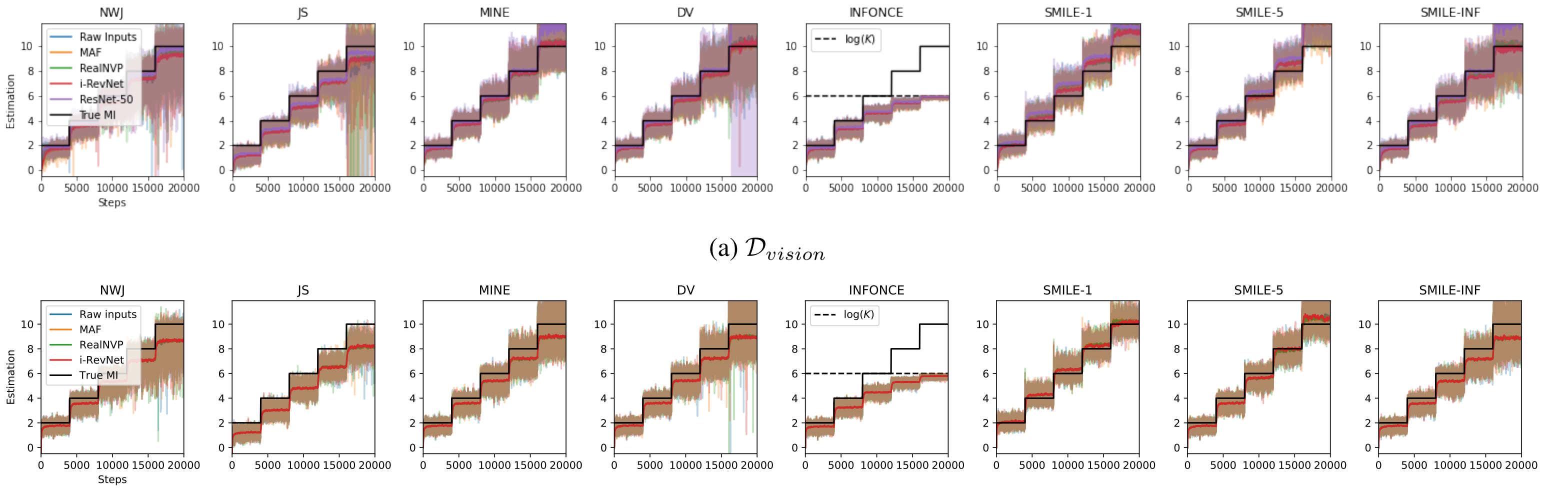
Figure 4.9. Estimation fails when too severe a nuisance is inserted. Although the nuisance does not affect the true MI value, the estimation fails when η is greater than 0.4.

4.3.6. Nuisance

Real-world datasets can contain various types of nuisance variables, and the nuisance variables do not share any information with the information source, following our definition in Section 4.1.2. For a quantitative study of the influence of nuisance variables for images, we conduct the experiments by increasing the nuisance intensity with the parameter $\eta \in [0, 1]$. The image x is written over the scaled background image $z \cdot \eta$. Thus, $\eta = 0$ corresponds to the image without nuisance (Figure 4.2(a-c)), and $\eta = 1$ corresponds to the image with the most severe nuisance (Figure 4.2(d)). Inserting nuisance does not affect the true MI values because we can still perfectly predict the class labels given the images. The results are summarized in Figure 4.9. Although the nuisance does not affect the true $I(X; Y)$, estimation fails when too severe a nuisance is inserted for all types of variational estimators.

4.3.7. Deep representations

In this section, we investigate the estimation accuracy of variational MI estimators in the case of deep representations (i.e., $I(g(X); g(Y))$ where $g(\cdot)$ is a deep network). We test three invertible networks, namely MAF (Papamakarios, Pavlakou, and Murray 2017), RealNVP (Dinh, Sohl-Dickstein, and Bengio 2016), and i-RevNet (Jacobsen, Smeulders, and Oyallon 2018) with random initialization. For the vision case, we additionally test a non-invertible network of ResNet-50 pre-trained based on the same dataset. As shown in Figure 4.10, estimation holds for the deep representations, regardless of which architecture is used.



(a) $\mathcal{D}_{vision}$

(b) $\mathcal{D}_{NLP}$

Figure 4.10. Estimation results for deep representations. For all types of encoder networks, we achieve accurate estimations regardless of the choice of network architecture.

4.4. Discussion: How can we make use of MI with practical datasets?

Our observation that a strong nuisance hinders the accurate estimation of MI restricts the usage of MI for analyzing the practical dataset and their representations, because nuisance variables would be inevitable in most cases. To overcome this problem, we suggest a method to train an additional encoder network $g(\cdot)$ and estimate MI for the learned representations. $g(\cdot)$ is trained to learn a representation $g(X)$ to predict the information source C from the given image X.

We first investigate the case of MNIST images with a background nuisance of CIFAR-10 with $\eta = 1$. We train $g(\cdot)$ to minimize the cross-entropy loss between the predictions and labels. After training, we estimate the MI between the learned representations of the penultimate layer of ResNet-50 as $\hat{I}(g(X); g(Y))$. The results are provided in Figure 4.11. Evidently, the estimations become accurate if we use the representations, rather than the raw inputs. In addition to the penultimate layer, we investigate the representations of all hidden layers of $g(\cdot)$. We denote the representations of the l-th layer of $g(\cdot)$ as $g_l(\cdot)$ and $l \in [1, L]$. Due to the data processing inequality, $I(g_L(X); g_L(Y)) \leq I(g_{L-1}(X); g_{L-1}(Y)) \leq \cdots \leq I(X; Y) = H(C)$ and the estimated values correspond to the lower bound of the true MI. As shown in Figure 4.12, the estimated MI values are significantly increased when the output size is changed, and tight estimation is available only for the top layers for all types of estimators. We attribute this result to learning task-specific features in the top layers (Kornblith et al. 2021), and it might be an interesting topic for future research.

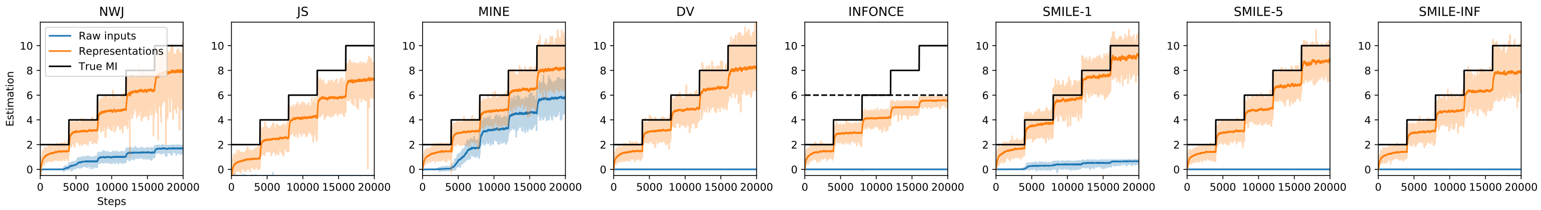
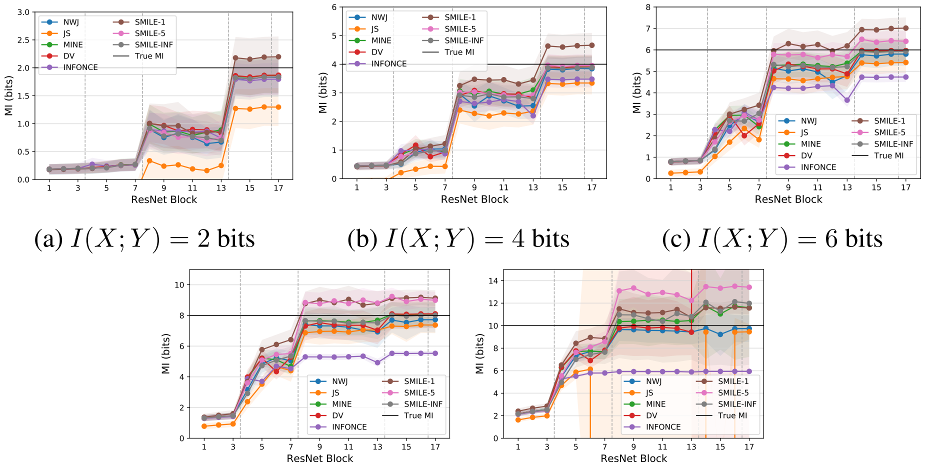


Figure 4.11. MI estimation results of the image dataset when a nuisance is inserted with $\eta = 1$. While we cannot achieve tight bounds for raw inputs, we achieve accurate estimations after the encoder network is trained for all types of estimators and all values of true MI.



(a) $I(X;Y) = 2$ bits

(b) $I(X;Y) = 4$ bits

(c) $I(X;Y) = 6$ bits

(d) $I(X;Y) = 8$ bits

(e) $I(X;Y) = 10$ bits

Figure 4.12. MI estimation results for hidden layers of ResNet-50. Dashed lines indicate boundaries between stages. We found that the upper layer representations retain the task-relevant information C. The transition clearly occurs when the output size changes. Further, the top layers provide the tight bound of the true MI while the lower layers do not.

Next, we consider the practical dataset, which does not satisfy our assumption that we need an error-free classification function $h_{\text{class}} : X \to C$, as in Theorem 2. If such a function does not exist, we can establish only the upper bound of true MI as $I(X;Y) \leq H(C)$, rather than its equality. In this case, we cannot access the exact ground-truth MI between two input variables, i.e., $I(X;Y)$. Instead, we can access the ground-truth MI between the representations, i.e., $I(g(X);g(Y))$, when we train the encoder network $g(\cdot)$ to guarantee the existence of an error-free classification function $h'_{\text{class}} : g(X) \to C$. We provide a more detailed explanation that the true MI is accessible when the encoder network g is trained to minimize the cross-entropy loss function between $g(X)$ and C as follows.

Theorem 2 has an assumption that we need an error-free classification function $h_{\text{class}} : X \to C$. If such a function does not exist, we can establish only the upper

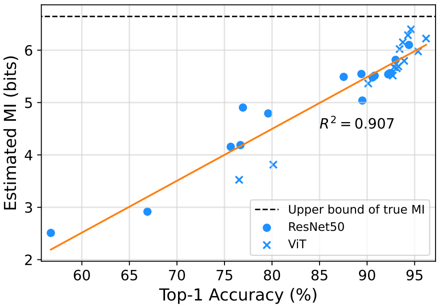


Figure 4.13. MI estimation results for ImageNet-100 dataset. We test 30 pre-trained models (16 for ResNet-50 and 14 for ViT). When the linear accuracy is close to 100%, we can achieve accurate estimation of the true MI.

bound of true MI as $I(X;Y) = H(C) - H(C|X) \leq H(C)$, instead of the equality. The proof is similar with the part of the upper bound in Chapter 3. If an error-free classification function h_{class} does not exist, we could minimize $H(C|X)$ by training an additional encoder network $g(\cdot)$ to minimize the cross entropy $H(C, \text{logit}(\hat{X}))$, where $\hat{X} = g(X)$.

$$\begin{aligned} H(C, \text{logit}(\hat{X})) &= H(C) + KL(C||\text{logit}(\hat{X})) \\ &= H(C) + KL(C||X) \\ &= H(C|X) + I(C;X) + KL(C||X) \\ &\geq H(C|X) \end{aligned}$$

The first equality comes from the deterministic property of g. When we train $g(\cdot)$ to have a sufficiently small cross-entropy loss, the conditional entropy $H(C|X)$ will be equally minimized. Finally, we establish the equality of $I(X;Y) = H(C)$ when we have a well-trained g, instead of h_{class}.

Finally, we test the practical dataset ImageNet-100. In this case, $\hat{I}(g(X); g(Y)) \leq I(g(X); g(Y)) \leq I(X;Y) \leq H(C) = \log 100 = 6.64$ bits. We use a variety of pre-trained models loaded from (Goyal et al. 2021; Khosla et al. 2020; Wightman 2019). We inspect 16 pre-trained ResNet-50 models and 14 pre-trained ViT models. All models are pre-trained with the ImageNet-1k dataset. We load the pre-trained models and evaluate the linear accuracy and InfoNCE estimator with a sufficiently large batch size during estimation. The results are shown in Figure 4.13. We cannot only achieve a tight estimation of the MI given by $I(X;Y) = H(C)$ when we have a sufficiently accurate encoder network $g(\cdot)$, but the estimated MI values are also highly correlated with the top-1 accuracy of deep representations ($R^2 = 0.907$).

Although our analysis framework requires an assumption of the existence of a perfect classifier h_{class}, this assumption is not necessary if we evaluate the representations based on the well-trained encoder network. Thus, MI estimated on the variational approaches can be an excellent metric for analyzing deep representations when we have a well-trained encoder network $g(\cdot)$.

4.5. Conclusion

In this chapter, we empirically examined the estimation accuracy of mutual information for a variety of scenarios. For a rigorous investigation with full availability of the true MI, number of information sources, representation dimension, and nuisance, we define a particular set of datasets for vision and NLP tasks. We found that several previous beliefs, including increasing that critic capacity is always beneficial for improving the estimation accuracy, should be reconsidered to be generalized across the data domains. Finally, we evaluated the estimation accuracy for practical datasets by training an encoder network for a targeted downstream task related to the information sources, i.e., the true joint distribution $p(x, y)$. In conclusion, it is necessary to access the true MI values when we analyze the estimated MI values.

Chapter 5. Examining Three Existing Beliefs on Mutual Information in Contrastive Learning

A long list of studies has been completed in the field of contrastive learning with mutual information. Some of the important topics that have been studied or implied can be listed as the following: large MI is necessary for learning useful representations (Oord, Li, and Vinyals 2018; Hjelm et al. 2018; Bachman, Hjelm, and Buchwalter 2019; Tschannen et al. 2019; Tian, Krishnan, and Isola 2020; Tian et al. 2020; Sordoni et al. 2021; Wu et al. 2020a); batch size (the number of negative samples) needs to be large because InfoNCE bound cannot estimate MI larger than $\mathcal{O}(\log K)$ where K is the batch size (Poole et al. 2019; McAllester and Stratos 2020; Chen et al. 2020b; Tian, Krishnan, and Isola 2020; Sordoni et al. 2021; Wu et al. 2020a; 2020b; Song and Ermon 2020); the optimal views should include the task-relevant information while discarding irrelevant information (Tian, Krishnan, and Isola 2020; Tian et al. 2020; Mitrovic et al. 2020; Tsai et al. 2020). While the previous studies are enlightening, we have also found that the investigation methods used in there can sometimes lead to misleading or incorrect conclusions. This observation has motivated us to develop the methods explained in Chapter 3 where the joint distribution $p(x, y)$ is carefully considered.

With the newly developed methods, we clarify how contrastive learning and mutual information are connected. To be specific, we investigate the three existing beliefs on mutual information in contrastive learning, focusing on the image classification as the downstream task. The below are the three beliefs that we investigate.

1. MI can measure how effective the representations are for the downstream task.

2. Small batch size limits mutual information and contrastive learning.

3. For designing an optimal view, we need to discard the task-irrelevant dependency

for a better generalization.

5.1. Method

We have summarized three key limitations to utilizing MI for analyzing the deep representations in Chapter 1: (1) the variational bounds of MI only provide the lower bound of MI, and it is not the true MI; (2) the choice of augmentation (positive pairing) determines the joint distribution $p(x, y)$, and the joint distribution determines the true MI; (3) we cannot regard the limitations for MI estimation same as the limitations for representation learning. In this study, we develop an analysis framework to overcome these limitations. We first suggest the same-class sampling to make use of true MI values and restrict the shared information between two views (corresponding to the limitation (1) and (2)). In addition, we introduce *CDP dataset* that always satisfies the assumption in Theorem 2. Thanks to the way CDP dataset is constructed, not only the exact $I_{\text{class}}(X; Y)$ is available, but also we can limit the shared and not to be shared information between two views (corresponding to the limitation (1) and (2)). Finally, we separate MI estimation into a post-training phase to overcome the limitation (3). We provide the detailed descriptions as following.

5.1.1. Post-training MI estimation

As explained in Section 2.1, InfoNCE can be used as a training loss or as a bound for MI estimation. Let's consider the training first. As shown in Figure 5.1(Top), training is not only dependent on the choice of loss but also on the choice of positive pairing $\mathcal{T}$. For brevity, we denote the loss as $\mathcal{L}_{\text{SimCLR}}$ and $\mathcal{L}_{\text{class}}$ when InfoNCE loss in Eq. (2.2) is used with $\mathcal{T}_{\text{SimCLR}}$ and $\mathcal{T}_{\text{class}}$, respectively. Because same-class sampling $\mathcal{T}_{\text{class}}$ requires class label, training with $\mathcal{L}_{\text{class}}$ implies a supervised training.

Second, let's consider MI estimation. Most, if not all, of the previous works have estimated MI during the training. This imposes a limitation where $\mathcal{T}$ for training and $\mathcal{T}$

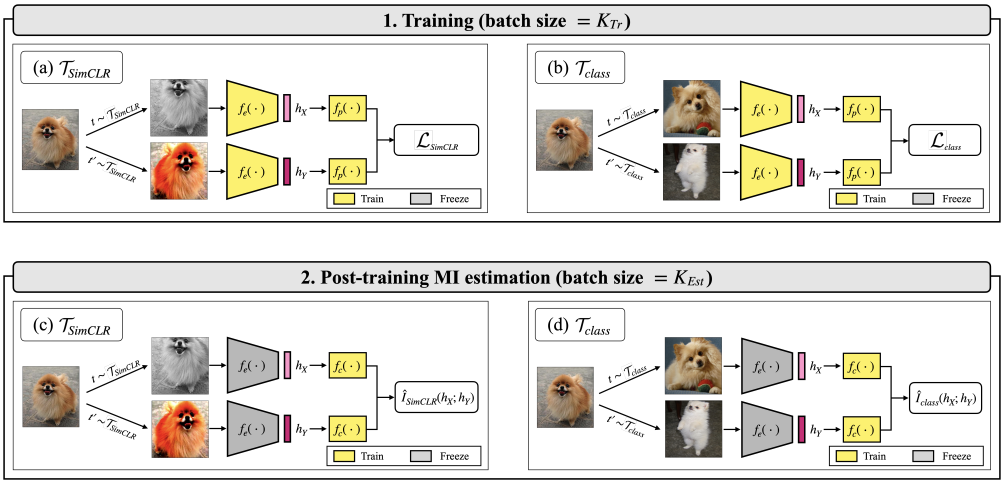
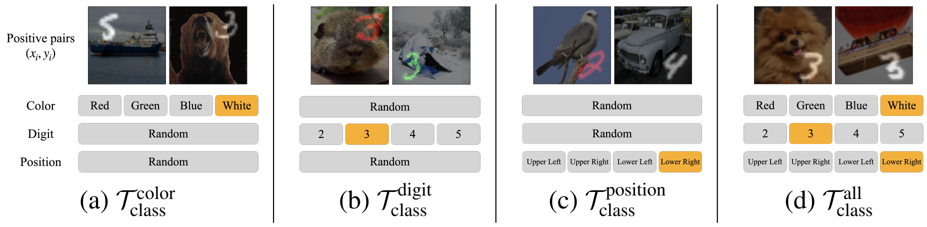

1. Training (batch size $= K_{Tr}$)
(a) $\mathcal{T}_{SimCLR}$
$t \sim \mathcal{T}_{SimCLR}$
$t' \sim \mathcal{T}_{SimCLR}$
$f_e(\cdot)$
h_X
$f_p(\cdot)$
$f_e(\cdot)$
h_Y
$f_p(\cdot)$
$\mathcal{L}_{SimCLR}$
Train
Freeze
(b) $\mathcal{T}_{class}$
$t \sim \mathcal{T}_{class}$
$t' \sim \mathcal{T}_{class}$
$f_e(\cdot)$
h_X
$f_p(\cdot)$
$f_e(\cdot)$
h_Y
$f_p(\cdot)$
$\mathcal{L}_{class}$
Train
Freeze
2. Post-training MI estimation (batch size $= K_{Est}$)
(c) $\mathcal{T}_{SimCLR}$
$t \sim \mathcal{T}_{SimCLR}$
$t' \sim \mathcal{T}_{SimCLR}$
$f_e(\cdot)$
h_X
$f_c(\cdot)$
$f_e(\cdot)$
h_Y
$f_c(\cdot)$
$\hat{I}_{SimCLR}(h_X; h_Y)$
Train
Freeze
(d) $\mathcal{T}_{class}$
$t \sim \mathcal{T}_{class}$
$t' \sim \mathcal{T}_{class}$
$f_e(\cdot)$
h_X
$f_c(\cdot)$
$f_e(\cdot)$
h_Y
$f_c(\cdot)$
$\hat{I}_{class}(h_X; h_Y)$
Train
Freeze

Figure 5.1. Training and MI estimation. (Top) Training: We train the encoder $f_e(\cdot)$ and the projection head $f_p(\cdot)$ to minimize the InfoNCE loss $\mathcal{L}$. (a) With unsupervised positive pairing $\mathcal{T}_{\text{SimCLR}}$. (b) With supervised positive pairing $\mathcal{T}_{\text{class}}$. (Bottom) Post-training MI estimation: We train the critic $f_c(\cdot)$ to maximize the InfoNCE bound $\hat{I}(h_X; h_Y)$ while $f_e(\cdot)$ is frozen. (c) With unsupervised positive pairing $\mathcal{T}_{\text{SimCLR}}$. (d) With supervised positive pairing $\mathcal{T}_{\text{class}}$.

for MI estimation cannot differ. Furthermore, the encoder weights are not fixed during training and thus the MI of a moving target needs to be estimated. To overcome the limitations, we propose *post-training MI estimation* that is illustrated in Figure 5.1(Bottom). With our post-training MI estimation, we have the flexibility to estimate MI that corresponds to any positive pairing and its joint distribution including $p_{\text{SimCLR}}(x, y)$ and $p_{\text{class}}(x, y)$. In other words, we can use $\mathcal{L}_{\text{SimCLR}}$ for training (i.e. generally used unsupervised contrastive learning) and estimate $\hat{I}_{\text{class}}(h_X; h_Y)$ for MI estimation. Also, we have the flexibility to choose any network pre-trained in a supervised or unsupervised way because the encoder network is kept frozen during the MI estimation phase. Overall, we can examine either $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ or $\hat{I}_{\text{class}}(h_X; h_Y)$ of any pre-trained network $f_e(\cdot)$ using the post-training MI estimation process shown in Figure 5.1(Bottom). Also, separating MI estimation into a post-training phase can improve the estimation accuracy



(a) $\mathcal{T}_{\text{class}}^{\text{color}}$ (b) $\mathcal{T}_{\text{class}}^{\text{digit}}$ (c) $\mathcal{T}_{\text{class}}^{\text{position}}$ (d) $\mathcal{T}_{\text{class}}^{\text{all}}$

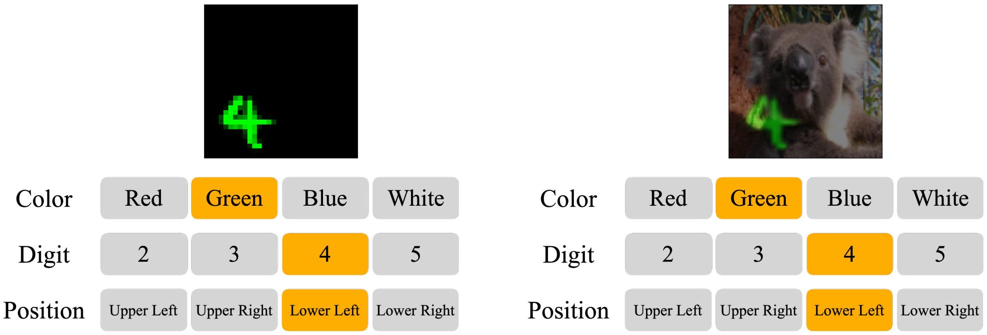
Figure 5.2. Manipulating true MI with CDP dataset. When only one of color, digit, and position is matched by same-class sampling as shown in (a), (b), and (c), the true MI is 2 bits ($I_{\text{class}}(X;Y) = 2$). When all three are consistently matched as shown in (d), the true MI is 6 bits ($I_{\text{class}}(X;Y) = 6$).

because we can use a larger batch size for the estimation without affecting the choice of batch size for training and the resulting learning dynamics of the encoder.

For training, a projection head $f_p(\cdot)$ is used as shown in Figure 5.1(Top). For MI estimation, a critic function $f_c(\cdot)$ is used as shown in Figure 5.1(Bottom). We use a common MLP network for both $f_p(\cdot)$ and $f_c(\cdot)$ to ensure a fair comparison. See Supplementary 5.2 for the details. Again, the introduction of supervised $\mathcal{L}_{\text{class}}$ is not for a practical purpose but only for in-depth investigations.

5.1.2. CDP dataset

In the existing MI analyses that are related to practical contrastive learning, only the estimated MI value has been studied simply because evaluating the true MI value has not been possible. For a dataset that allow the class label to be clearly identified for each image, however, the true MI value for same-class sampling can be proven to be equal to the class label entropy, $H(C)$. The proof is provided in Chapter 3. To take advantage of this special case, we introduce a synthetic dataset named *CDP dataset*. In CDP dataset, each image is constructed by uniformly choosing a color c_{color} from $\{\text{Red}, \text{Green}, \text{Blue}, \text{White}\}$, a digit c_{digit} from $\{2, 3, 4, 5\}$, and a position c_{position} from $\{\text{Upper left}, \text{Upper right}, \text{Lower left}, \text{Lower right}\}$. The three attributes are independently chosen for each image. Because of the uniform selection, the entropy of each class label is clearly $H(C_{\text{color}}) = H(C_{\text{digit}}) = H(C_{\text{position}}) = 2$ bits. Note that random ImageNet examples are inserted in the background to make the dataset realistic.



(a) Without background nuisance

(b) With background nuisance

Figure 5.3. An example of how to generate a CDP dataset. Detailed descriptions are provided in the text.

For a better understanding, we provide an example of how to generate a CDP dataset in Figure 5.3. We first define the label variable c_i as the combination of three independent attributes. Following $c_i = (\text{Green}, 4, \text{Lower Left})$, we generate the image as described in Figure 5.3(a). Note that we use the digit images from the MNIST dataset after resizing and coloring. Obviously, there exists an error-free classification function f_{class} which predicts the label information from the given image. In addition to these plain images, we generate a more complex and realistic version of the CDP dataset, still satisfying the assumption of f_{class}. To this end, we insert the randomly chosen background image from ImageNet as shown in Figure 5.3(b). To satisfy the assumption of f_{class}, we make the source images of Figure 5.3(a) on top without any occlusion. Thus, we still guarantee that f_{class} exists after we insert the background nuisance.

(Tian et al. 2020; Hermann and Lampinen 2020; Chen, Luo, and Li 2021) have suggested similar datasets but they focused on feature suppression or task-dependence of optimal views, not mutual information. The RandBit dataset of (Chen, Luo, and Li 2021) is also similar, but it is far from practical images and provides only a loose bound of MI. We also note that CDP dataset does not have object-centric bias addressed in

(Purushwalkam and Gupta 2020) and can be easily controlled by enforcing dependencies among the three information sources.

Thanks to the way the CDP dataset is constructed, the true MI under same-class sampling can be easily manipulated as shown in Figure 5.2. If only the color attribute is consistently chosen for each pair (Figure 5.2(a)), it corresponds to a downstream task whose class label is the color information and the positive pairing is denoted as $\mathcal{T}_{\text{class}}^{\text{color}}$. In this case, the true MI is $I_{\text{class}}(X;Y) = H(C_{\text{color}}) = 2$ bits. Similarly, $I_{\text{class}}(X;Y) = 2$ bits for Figure 5.2(b) and Figure 5.2(c). When all three attributes are consistently chosen for each pair (Figure 5.2(d)), it corresponds to a downstream task whose class label is the combination of color, digit, and position information. Then, the true MI is $I_{\text{class}}(X;Y) = H(C_{\text{color}}) + H(C_{\text{digit}}) + H(C_{\text{position}}) = 6$ bits. Note that the entropies add up because looking at one of the pair provides the exact information of the color, digit, and position of the other image.

Detailed example of the same-class sampling for CDP dataset: Because both the same-class sampling and the CDP dataset are suggested for the first time in this study, we provide a more detailed example of the same-class sampling for CDP dataset. For a convenience, we fix the image x_i as in Figure 5.3(b) and sample the y_i depending on the different label information for same-class sampling. Let's start with the $\mathcal{T}_{\text{class}}^{\text{color}}$, i.e. same-class sampling based on C_{color}. As described in Figure 5.5(a), the positive pair $(x_i, y_i) \sim p(x, y)$ shares the color information only and the other labels are determined independently. Obviously, there is no reason for the representations to learn the invariance for digit or position. In a similar way, $\mathcal{T}_{\text{class}}^{\text{digit}}$ (same-class sampling based on C_{digit}, Figure 5.5(b)) and $\mathcal{T}_{\text{class}}^{\text{position}}$ (same-class sampling based on C_{position}, Figure 5.5(c)) enforce the positive pair to share the targeted label information only. Finally, for $\mathcal{T}_{\text{class}}^{\text{all}} = \mathcal{T}_{\text{class}}$ (same-class sampling based on C_{all}), the positive pair $(x_i, y_i) \sim p(x, y)$ shares all the three attributes as shown in Figure 5.5(d). Note that the background images are always selected to be different for the positive pairs and they do not affect the MI value.

CDP dataset vs. Practical dataset: We summarize the same-class sampling for CDP dataset and the practical dataset as an example of ImageNet in Figure 5.4. CDP dataset satisfies the assumption that an error-free classification function f_{class} exists. Thus, the true MI value is accessible as $I_{\text{class}}(X;Y) = H(C)$. On the other hand, it is not trivial to guarantee that the practical dataset satisfies the assumption that an error-free classification function f_{class} exists. Thus, the upper bound of $I_{\text{class}}(X;Y)$ is only available as $H(C)$. Throughout our study, we use the CDP dataset when we need a true MI value for the analysis, and we use the various practical dataset in addition to the CDP dataset otherwise.

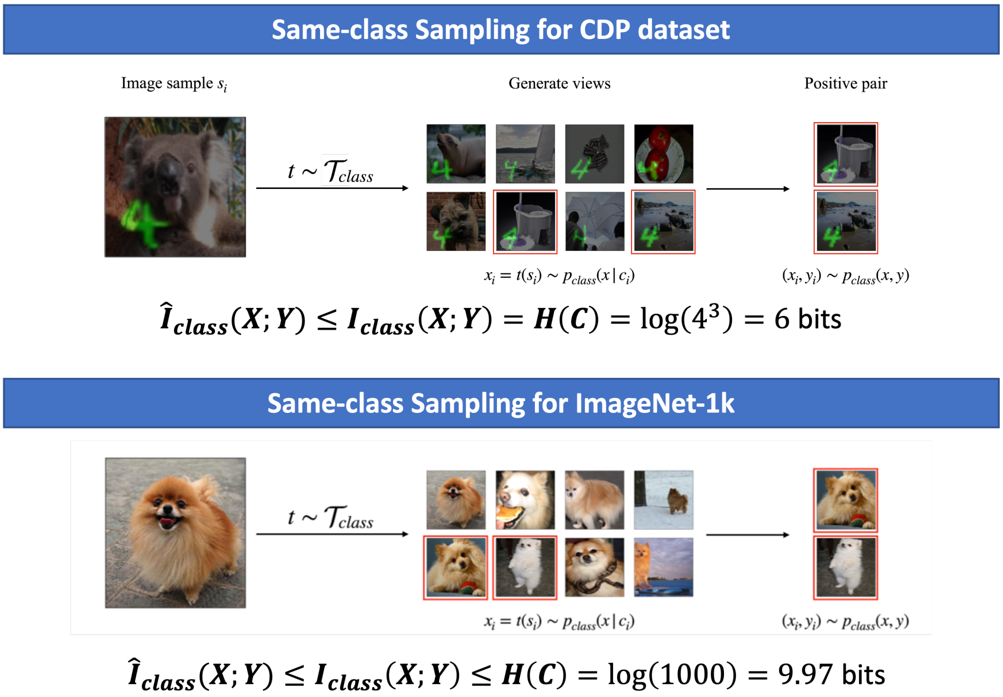
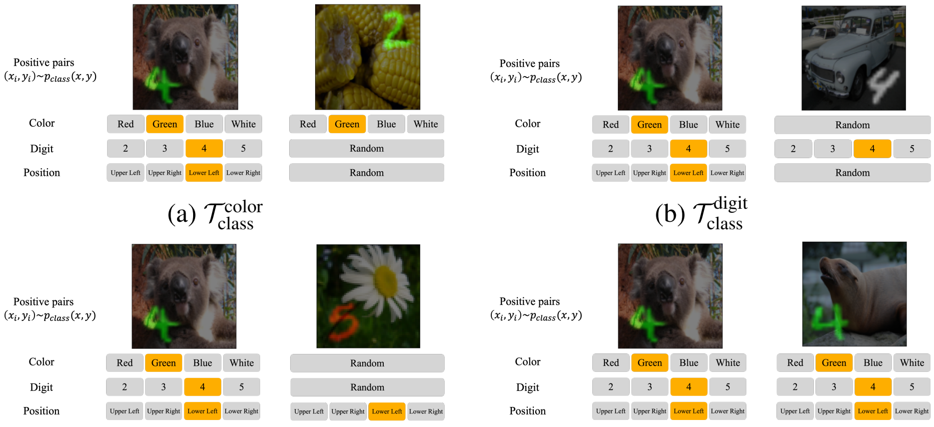

Same-class Sampling for CDP dataset
Image sample s_i
Generate views
Positive pair
$t \sim \mathcal{T}_{class}$
$x_i = t(s_i) \sim p_{class}(x \mid c_i)$
$(x_i, y_i) \sim p_{class}(x, y)$
$\hat{I}_{class}(X;Y) \leq I_{class}(X;Y) = H(C) = \log(4^3) = 6$ bits
Same-class Sampling for ImageNet-1k
$t \sim \mathcal{T}_{class}$
$x_i = t(s_i) \sim p_{class}(x \mid c_i)$
$(x_i, y_i) \sim p_{class}(x, y)$
$\hat{I}_{class}(X;Y) \leq I_{class}(X;Y) \leq H(C) = \log(1000) = 9.97$ bits

Figure 5.4. An example of applying same-class sampling for the CDP dataset and the practical dataset (ImageNet-1k).



(a) $\mathcal{T}_{\text{class}}^{\text{color}}$ (b) $\mathcal{T}_{\text{class}}^{\text{digit}}$

(c) $\mathcal{T}_{\text{class}}^{\text{position}}$ (d) $\mathcal{T}_{\text{class}}^{\text{all}}$

Figure 5.5. An example of same-class sampling for CDP dataset.

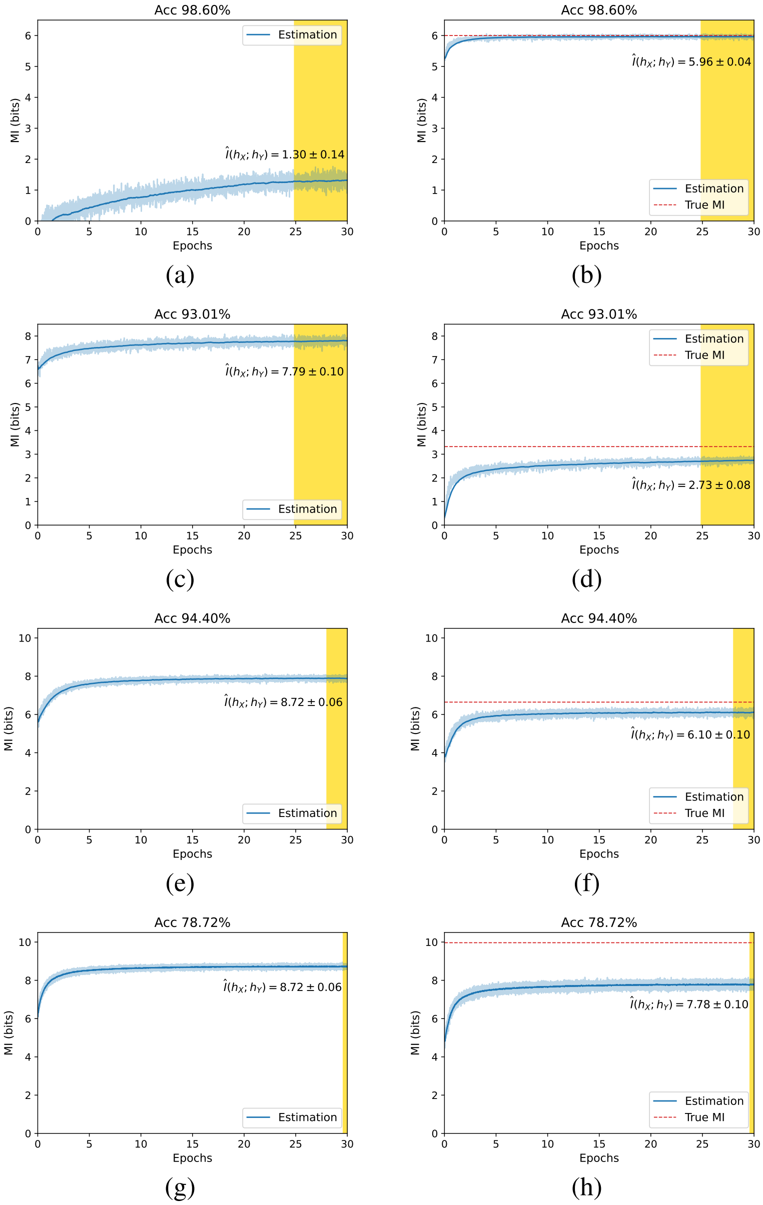
5.2. Experimental setups

5.2.1. Training

In our study, we train the encoder $f_e(\cdot)$ of ResNet-18 and ResNet-50 and the projection head $f_p(\cdot)$ of 2-layer MLP with batch normalization for 100 epochs. We set the batch size K_{Tr} as 256 for CDP and CIFAR-10, and 128 for ImageNet-100 and ImageNet-1k. We set the temperature scalar τ as 0.5 for CIFAR-10 and 0.2 for other datasets. We optimize the InfoNCE loss using SGD with learning rate of 0.001 and weight decay of $1e^{-4}$ for CDP and CIFAR-10, and with learning rate of 0.4 and weight decay of 0.00002 for ImageNet. We also use linear warm-up for the first 3 epochs (10 for ImageNet), and decay the learning rate with the cosine decay schedule without restarts (Loshchilov and Hutter 2016; Goyal et al. 2017). We carried out all the experiments using PyTorch on a single Nvidia RTX 3090 GPU.

5.2.2. Post-training MI estimation

The critic $f_c(\cdot)$ can be flexibly chosen as explained in (Poole et al. 2019; Song and Ermon 2019), but we set it identical in architecture and hyperparameters as the projection head $f_p(\cdot)$ of the training stage. The estimation is performed with the epoch size of 30. We have chosen the epoch size based on the learning curves of a variety of post-training MI estimation results shown in Figure 5.6. We empirically found that 30 is sufficiently large for the estimations to converge. MI estimation aims to maximize the lower bound of MI, and we define the final estimated MI as the average of the last 1000 steps (as highlighted in the figures) to deal with the estimation variance. To prevent the $\log(2K_{\text{Est}} - 1)$ becoming a limiting factor of the MI estimation, we have chosen the MI estimation batch size K_{Est} to be sufficiently large. We set K_{Est} as 256 for CDP and CIFAR-10 and 512 for ImageNet-100 and ImageNet-1k. Note that K_{Est} is independently chosen from K_{Tr}, the batch size of training. Unlike the training stage, MI estimation is not affected by the temperature scalar τ, and we set $\tau = 0.1$ throughout our study.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 5.6. Examples of post-training MI estimation: (a) CDP, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (b) CDP, $\hat{I}_{\text{class}}(h_X; h_Y)$, (c) CIFAR-10, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (d) CIFAR-10, $\hat{I}_{\text{class}}(h_X; h_Y)$, (e) ImageNet-100, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (f) ImageNet-100, $\hat{I}_{\text{class}}(h_X; h_Y)$, (g) ImageNet-1k, $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$, (h) ImageNet-1k, $\hat{I}_{\text{class}}(h_X; h_Y)$. Note that the MI estimation $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ in (a) is relatively smaller when compared to the $\hat{I}_{\text{class}}(h_X; h_Y)$ in (b). This is an example where $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ does not properly reflect the downstream task performance while $\hat{I}_{\text{class}}(h_X; h_Y)$ does.

5.3. Results

5.3.1. A small batch size is a limiting factor for MI estimation but not for contrastive learning.

Existing belief 1:

A small batch size is undesirable for contrastive learning because of InfoNCE's $\mathcal{O}(\log K)$ bound (Hjelm et al. 2018; Tian, Krishnan, and Isola 2020; Bachman, Hjelm, and Buchwalter 2019; Chen et al. 2020b; Sordoni et al. 2021; Wu et al. 2020a; Song and Ermon 2020).

Correction 1:

A small batch size limits the training loss, but it limits neither the information in the learned representation nor the downstream-task performance.

It is a well-known fact that the estimated MI in Eq. (2.3) is upper bounded by $\log(2K_{\text{Tr}} - 1)$ (Oord, Li, and Vinyals 2018; Sordoni et al. 2021; McAllester and Stratos 2020; Poole et al. 2019), where K_{Tr} is the batch size of training. See Section 2.1 for the derivation. Because of the bound, it has been often believed that a small batch size affects the contrastive learning negatively. To overcome this limitation, many of the previous works have increased the batch size (Hjelm et al. 2018; Tian, Krishnan, and Isola 2020; Bachman, Hjelm, and Buchwalter 2019) or have modified the InfoNCE loss (Sordoni et al. 2021; Wu et al. 2020a; Song and Ermon 2020). The existing works, however, have estimated MI concurrently during the training phase.

To examine whether the existing belief is always true and there is no counter-example, we have performed experiments as summarised in Figure 5.7. Here we need the true MI value, so we use CDP dataset and $\hat{I}_{\text{class}}(h_X; h_Y)$. For training, we decrease the batch size K_{Tr} from 256 to 2. For MI estimation, we fix the batch size K_{Est} as 256 to make sure we can estimate the true MI of 6 bits. If the existing belief is correct,

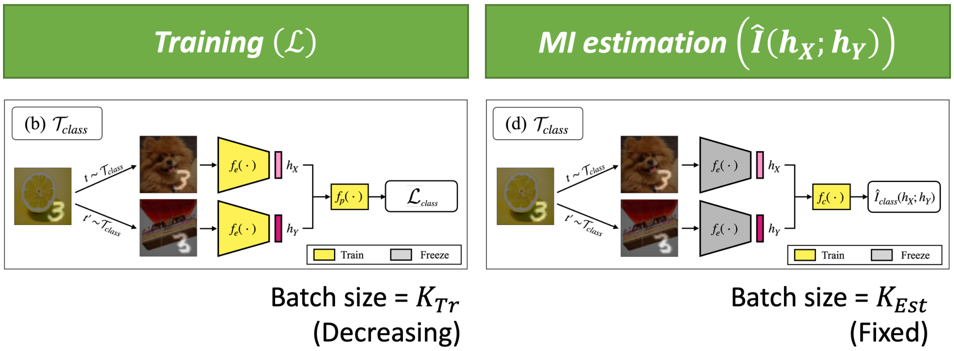

Training $(\mathcal{L})$
MI estimation $\left(\hat{I}(h_X; h_Y)\right)$
(b) $\mathcal{T}_{class}$
$t \sim \mathcal{T}_{class}$
$t' \sim \mathcal{T}_{class}$
$f_e(\cdot)$
h_X
h_Y
$f_p(\cdot)$
$\mathcal{L}_{class}$
Train
Freeze
Batch size = K_{Tr}
(Decreasing)
(d) $\mathcal{T}_{class}$
$f_c(\cdot)$
$\hat{I}_{class}(h_X; h_Y)$
Batch size = K_{Est}
(Fixed)

Figure 5.7. A summary of experimental setups. For both phases, we use same-class sampling for positive pairing with CDP dataset. For training, we utilize the batch size K_{Tr} from 2 to 256. For post-training MI estimation, we fix the batch size K_{Est} as 256.

the result can be described in Figure 5.8. In other words, we cannot achieve a high downstream-task accuracy for a small K_{Tr} and the learned representations cannot share the information larger than $\mathcal{O}(\log K)$.

We provide the results when we have performed two sets of experiments in Figure 5.9. Even though the estimated MI with the training loss is limited by $\log{(2K_{\text{Tr}} - 1)}$, we can see that the post-training MI estimation is almost the same as the true MI $(= 6$ bits) and that the performance is over 96% for all the cases. Clearly, $\log{(2K_{\text{Tr}} - 1)}$ bound is not necessarily harmful and a small batch size does not limit the representation learning. We also note that $\hat{I}_{\text{class}}(h_X; h_Y)$ is almost identical to the ground-truth MI, i.e., $\hat{I}_{\text{class}}(h_X; h_Y) \approx$ 6bits $= H(C)$. Thus, this result supports that the CDP dataset satisfies $I_{\text{class}}(X; Y) = H(C)$.

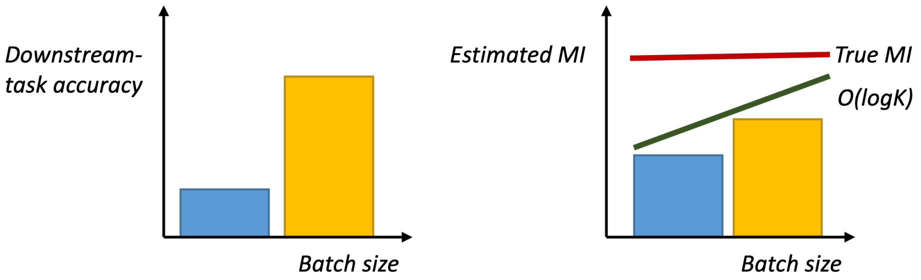
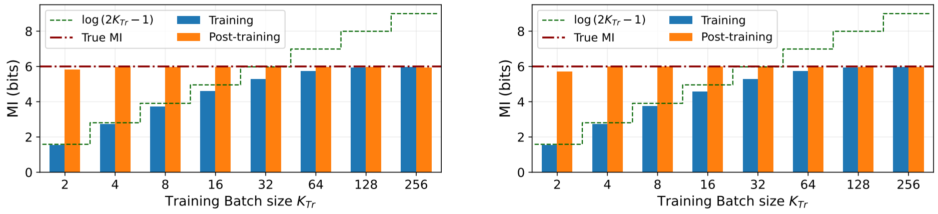


Figure 5.8. If the existing belief is correct, the results should be as above. (Left) The downstream-task accuracy should be improved when we increase the training batch size. (Right) We cannot estimate the MI larger than $\mathcal{O}(\log K)$.

K_{Tr}	2	4	8	16	32	64	128	256
Acc. (%)	98.8	**99.8**	99.7	99.7	99.6	99.4	99.4	99.2

K_{Tr}	2	4	8	16	32	64	128	256
Acc. (%)	96.0	**99.7**	**99.7**	**99.7**	99.5	99.2	99.0	98.6



(a) ResNet-18

(b) ResNet-50

Figure 5.9. True MI, training MI (conventional estimating of MI at the time of training using the training loss), and our post-training MI. For CDP dataset, we train two ResNet models using $\hat{I}_{\text{class}}(h_X; h_Y)$ as the loss (as in Figure 5.1(b)). We evaluate $\hat{I}_{\text{class}}(h_X; h_Y)$ at the end of training (blue) and post-training (orange). During training, the MI is upper bounded by $\log{(2K_{\text{Tr}}-1)}$ (dashed lines of green color). After the training is complete, the network is frozen and we evaluate the MI using a large batch size of $K_{\text{Est}}=256$. Even though the training MI is limited by the $\log{(2K_{\text{Tr}}-1)}$ bound, the post-training MI turns out to be almost the same as the true MI ($=6$ bits). Obviously, the trained model can represent sufficiently large amount of information.

5.3.2. Augmentation-based MI and other metrics are not effective, but $\hat{I}_{\text{class}}(h_X; h_Y)$ is effective.

Existing belief 2:

- MI *cannot* measure how effective the representation is for the downstream task's performance (Tschannen et al. 2019).
- Instead, other metrics such as uniformity (Wang and Isola 2020; Wang and Liu 2021), alignment (Wang and Isola 2020), tolerance (Wang and Liu 2021), and linear CKA (Nguyen, Raghu, and Kornblith 2020; Song et al. 2012; Nguyen, Raghu, and Kornblith 2022) are more relevant and useful than MI.

Correction 2:

The only metric (among the metrics that we have investigated) that is strongly relevant to the downstream-task performance is the MI of the downstream-task information itself.

The early contrastive learning studies (Oord, Li, and Vinyals 2018; Hjelm et al. 2018; Bachman, Hjelm, and Buchwalter 2019; Sordoni et al. 2021; Tian, Krishnan, and Isola 2020) have regarded the minimization of InfoNCE loss to be equivalent to the maximization of MI. The existing belief in Section 5.3.1 is an example. Then, (Tschannen et al. 2019) empirically showed that the estimated MI does not correlate well with the downstream-task performance. The analysis method in the work, however, was not rigorous in that only a particular choice of augmentation and the corresponding joint distribution $p_{\text{aug}}(x, y)$ were studied. Without addressing exactly what information is shared by $p_{\text{aug}}(x, y)$, the analysis can be quite misleading.

Subsequent works have suggested a variety of metrics to evaluate and explain the representation quality. Well-known metrics include alignment (Wang and Isola 2020), uniformity (Wang and Isola 2020; Wang and Liu 2021), tolerance (Wang and Liu 2021), and linear CKA (Nguyen, Raghu, and Kornblith 2020; Song et al. 2012;

Nguyen, Raghu, and Kornblith 2022). While the suggested metrics have become popular because they are intuitive and enlightening, there has been no attempt to provide a comprehensive analysis on how reliable the metrics are. We describe the metrics as below.

Representation evaluation metrics

The metrics are summarized below. For the implementation, we either adopt the authors' code (Wang and Isola 2020) or implement it by ourselves based on the equations in the paper (Wang and Liu 2021; Nguyen, Raghu, and Kornblith 2020).

- Alignment (Wang and Isola 2020): expected distance between positive pairs defined by $\mathcal{T}_{\text{aug}}$. Two views of positive pair should be mapped to nearby features, and thus be (mostly) invariant to unneeded noise factors. Representations are more aligned when the metric is smaller.

$$\text{Alignment} = \mathbb{E}_{(x,y)\sim p_{\text{pos}}}\left[||h_x - h_y||_2^{\alpha}\right]$$

- Uniformity (Wang and Isola 2020): the logarithm of the average pairwise Gaussian potential. Feature vectors should be roughly uniformly distributed on the unit hypersphere, preserving as much information of the data as possible. Representations are more uniform when the metric is smaller.

$$\text{Uniformity} = \log \mathbb{E}_{x,y\sim p_{\text{data}}}\left[e^{-t||h_x - h_y||_2^2}\right]$$

- Tolerance (Wang and Liu 2021): mean similarity of samples of the same class. It utilize the supervised information. Representations are more tolerant when the metric is higher.

$$\text{Tolerance} = \mathbb{E}_{x,y\sim p_{\text{data}}}\left[(h_x^T h_y)) \cdot \mathbb{1}_{c_x = c_y}\right]$$

- Linear CKA (Centered Kernel Alignment) (Nguyen, Raghu, and Kornblith 2020; Song et al. 2012; Nguyen, Raghu, and Kornblith 2022): the similarity between pairs of representations. We adopt the minibatch estimators and set the batch size as 200. Representations are more similar when the metric is higher. It is defined as

$$\text{Linear CKA} = \frac{1}{n(n-3)}\left(\text{tr}(\tilde{K}\tilde{L}) + \frac{\mathbf{1}^T\tilde{K}\mathbf{1}\mathbf{1}^T\tilde{L}\mathbf{1}}{(n-1)(n-2)} - \frac{2}{n-2}\mathbf{1}^T\tilde{K}\tilde{L}\mathbf{1}\right),$$

where $K = XX^T$, $L = YY^T$, $\tilde{K}$ and $\tilde{L}$ are obtained by setting the diagonal entries of K and L to zero, and X and Y denote the representation matrix for each view. This metric is not directly used to evaluate contrastive learning, and we assume $(x, y) \sim p_{\text{class}}(x, y)$. Therefore, it also utilizes the supervised information.

To investigate the existing beliefs, we have designed an experiment where the representations of many pre-trained networks can be carefully compared. To better understand the existing beliefs, we have followed the previous works and examined the relationship between each metric and the downstream-task performance. The first experiment's results can be found in Table 5.1. By examining Pearson's correlation and Kendall's rank correlation, the conclusion by (Tschannen et al. 2019) can be confirmed for $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$. For $\hat{I}_{\text{class}}(h_X; h_Y)$ whose joint distribution is directly related to the downstream task's class label information, however, the MI correlates very well with the downstream-task performance. Therefore, we can see that it is misleading to say that MI in general does not correlate well with the downstream-task performance. Clearly, $\hat{I}_{\text{class}}(h_X; h_Y)$, the MI that is directly associated with the downstream task's class label information, correlates with the downstream-task performance very well.

The experiment was repeated for five other scenarios, and the summary of Pearson's correlation results can be found in Table 5.3. In the table, we are also showing the results for the other metrics. Surprisingly, none of the known metrics shows a high correlation. The only metric that consistently shows a high correlation is the $\hat{I}_{\text{class}}(h_X; h_Y)$, implying that the downstream-task information itself (i.e. class label information) is the only metric that correlates well with the downstream-task performance. Note that the

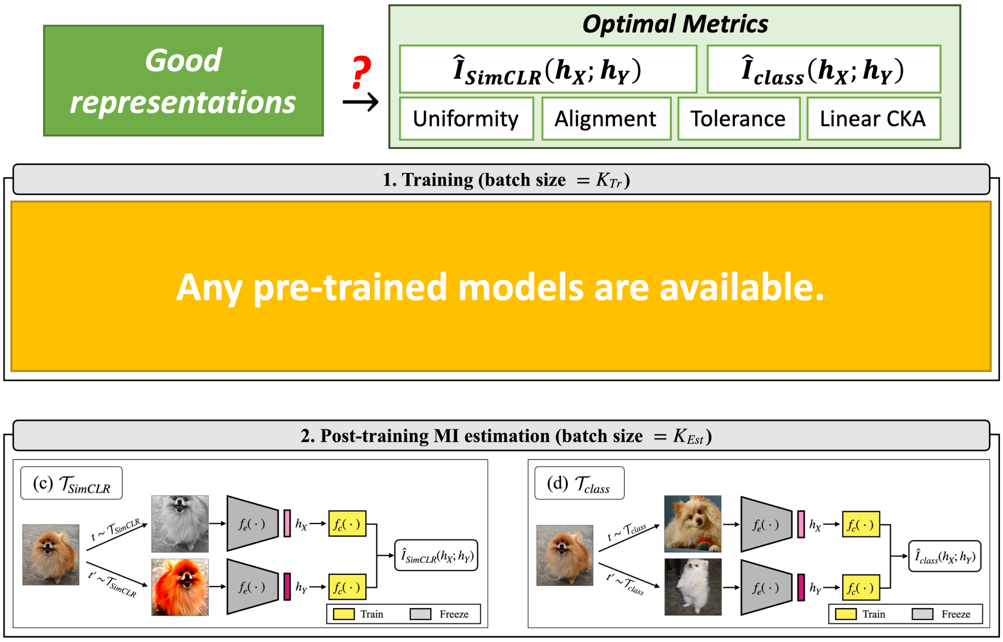

Good representations
?
Optimal Metrics
$\hat{I}_{SimCLR}(h_X; h_Y)$
$\hat{I}_{class}(h_X; h_Y)$
Uniformity
Alignment
Tolerance
Linear CKA
1. Training (batch size $= K_{Tr}$)
Any pre-trained models are available.
2. Post-training MI estimation (batch size $= K_{Est}$)
(c) $\mathcal{T}_{SimCLR}$
$t \sim \mathcal{T}_{SimCLR}$
$t' \sim \mathcal{T}_{SimCLR}$
$f_e(\cdot)$
h_X
h_Y
$f_c(\cdot)$
$\hat{I}_{SimCLR}(h_X; h_Y)$
Train
Freeze
(d) $\mathcal{T}_{class}$
$t \sim \mathcal{T}_{class}$
$t' \sim \mathcal{T}_{class}$
$\hat{I}_{class}(h_X; h_Y)$

Figure 5.10. A summary of experimental setups. To examine the good representations has the optimal metric, we investigate six representation metrics. We do not need to make use of the exact value of true MI, so we use various practical datasets, including CIFAR-10 and ImageNet. Also, we do not need to consider how the encoder network is trained, so we use lots of various pre-trained models. Then, we evaluate each metric in post-training phase.

class label information is also utilized by tolerance and linear CKA. So, they are also supervised metrics like $\hat{I}_{\text{class}}(h_X; h_Y)$, but they fail to achieve a high correlation. The full experimental results of all the scenarios can be found in Supplementary A.2.

Table 5.1. Post-training MI estimation results for ResNet-50 on ImageNet-100 and ImageNet-1k. Sixteen pre-trained models are used to evaluate the effectiveness of $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ and $\hat{I}_{\text{class}}(h_X; h_Y)$.

Algorithm	*ImageNet-100*			*ImageNet-1k*		
	Acc. (%)	$\hat{I}_{\text{SimCLR}}(h_X; h_Y)$	$\hat{I}_{\text{class}}(h_X; h_Y)$	Acc. (%)	$\hat{I}_{\text{SimCLR}}(h_X; h_Y)$	$\hat{I}_{\text{class}}(h_X; h_Y)$
SupCon (Khosla et al. 2020)	94.40	7.889	6.100	78.72	8.722	7.783
Supervised pretrained	93.00	7.598	5.816	74.11	8.378	6.761
SwAV (Caron et al. 2020)	92.52	8.541	5.560	74.78	9.428	6.214
DeepCluster-v2 (Caron et al. 2020)	92.38	8.540	5.559	73.65	9.416	6.232
DINO (Caron et al. 2021)	92.22	8.443	5.539	74.22	9.313	6.133
Barlow Twins (Zbontar et al. 2021)	90.80	8.528	5.513	72.82	9.407	6.157
PIRL (Misra and Maaten 2020)	90.58	8.584	5.480	70.51	9.481	6.247
SeLa-v2 (Caron et al. 2020)	89.50	6.020	5.039	69.66	7.354	5.774
SimCLR (Chen et al. 2020b)	89.40	8.669	5.546	69.12	9.580	6.277
MoCo-v2 (Chen et al. 2020c)	87.54	8.592	5.490	63.89	9.499	6.221
NPID++ (Misra and Maaten 2020)	79.60	8.190	4.792	56.60	9.009	4.692
MoCo (He et al. 2020)	76.94	8.338	4.904	47.05	9.155	4.907
NPID (Wu et al. 2018)	76.68	8.039	4.188	52.70	8.821	3.836
ClusterFit (Yan et al. 2020)	75.66	8.016	4.155	48.81	8.773	3.915
RotNet (Gidaris, Singh, and Komodakis 2018)	66.90	7.020	2.916	41.54	7.696	2.802
Jigsaw (Noroozi and Favaro 2016)	56.74	6.339	2.510	30.85	7.155	2.583
Pearson's ρ with Acc.		0.510	0.967		0.535	0.943
Kendall's τ_K with Acc.		0.233	0.883		0.233	0.617

Table 5.2. Post-training MI estimation results for ViT on ImageNet-100 and ImageNet-1k. Fourteen pre-trained models are used to evaluate the effectiveness of $\hat{I}_{\text{SimCLR}}(h_X; h_Y)$ and $\hat{I}_{\text{class}}(h_X; h_Y)$. Because of the computational budge, we exclude the two largest models for ImageNet-1k.

Algorithm	*ImageNet-100*			*ImageNet-1k*		
	Acc. (%)	$\hat{I}_{\text{SimCLR}}(h_X; h_Y)$	$\hat{I}_{\text{class}}(h_X; h_Y)$	Acc. (%)	$\hat{I}_{\text{SimCLR}}(h_X; h_Y)$	$\hat{I}_{\text{class}}(h_X; h_Y)$
Swin-B (Liu et al. 2021)	96.20	8.073	6.222	-	-	-
Supervised pretrained (ViT-B/16) (Dosovitskiy et al. 2021)	95.36	8.252	5.977	78.93	9.199	7.208
PiT-B (Heo et al. 2021)	94.62	7.895	6.398	-	-	-
DeiT (ViT-B/16) (Touvron et al. 2021a)	94.30	7.799	6.287	78.34	8.679	8.009
CaiT (XXS-36/16) (Touvron et al. 2021b)	93.90	7.492	5.795	75.67	8.373	6.795
PiT-S (Heo et al. 2021)	94.62	7.895	6.398	76.81	8.513	7.543
DeiT (ViT-S/16) (Touvron et al. 2021a)	93.42	7.435	6.021	75.59	8.278	7.280
CaiT (XXS-24/16) (Touvron et al. 2021b)	93.28	7.488	5.690	74.09	8.315	6.547
MoCo(v3) (ViT-B/16) (Chen, Xie, and He 2021)	93.12	8.594	5.654	75.51	9.524	6.658
DINO (ViT-B/16) (Caron et al. 2021)	92.84	8.454	5.675	73.28	9.367	6.598
Supervised pretrained (ViT-S/16) (Dosovitskiy et al. 2021)	92.70	6.863	5.515	72.85	7.572	6.233
DeiT (ViT-T/16) (Touvron et al. 2021a)	90.12	7.186	5.365	68.67	7.874	5.883
Supervised pretrained (ViT-T/16) (Dosovitskiy et al. 2021)	80.14	4.988	3.814	53.01	5.474	3.741
DINO (ViT-S/16) (Caron et al. 2021)	76.54	6.868	3.525	51.11	7.426	3.316
Pearson's ρ with Acc.		0.721	0.974		0.783	0.977
Kendall's τ_K with Acc.		0.516	0.802		0.576	0.848

Table 5.3. Summary of seven experiments. Except for $\hat{I}_{\text{class}}(h_X; h_Y)$ that directly utilizes downstream class information in $\mathcal{T}_{\text{class}}$, all the other known metrics turn out to be ineffective for assessing downstream task performance. In the case of alignment and uniformity, smaller values indicate better representations, so we flipped the signs. Note that the class label information is also utilized by tolerance and linear CKA.

Encoder	Dataset	Metrics					
		Alignment	Uniformity	Tolerance	Linear CKA	$\hat{I}_{\text{SimCLR}}(h_X; h_Y)$	$\hat{I}_{\text{class}}(h_X; h_Y)$
Pearson's Correlation Coefficient ρ with linear accuracy							
ResNet-$\{18, 50\}$	CDP	-0.977	-0.058	0.956	**0.992**	-0.988	0.990
ResNet-$\{18, 50\}$	CIFAR-10	-0.738	-0.319	0.121	-0.503	-0.041	**0.634**
ResNet-$\{18, 50\}$	ImageNet-100	0.165	-0.197	0.214	0.410	0.085	**0.805**
ResNet-50(Pretrained)	ImageNet-100	0.286	0.265	-0.227	0.722	0.510	**0.967**
ResNet-50(Pretrained)	ImageNet-1k	0.175	0.157	-0.132	0.451	0.535	**0.943**
ViT(Pretrained)	ImageNet-100	-0.102	0.623	-0.395	0.856	0.721	**0.974**
ViT(Pretrained)	ImageNet-1k	-0.077	0.561	-0.392	0.203	0.783	**0.977**
Average		-0.181	0.147	0.021	0.447	0.229	**0.899**
Kendall's Rank Correlation Coefficient τ_K with linear accuracy							
ResNet-$\{18, 50\}$	CDP	-0.545	0.061	0.485	0.333	-0.727	**0.545**
ResNet-$\{18, 50\}$	CIFAR-10	-0.600	-0.067	0.333	-0.467	-0.067	**0.467**
ResNet-$\{18, 50\}$	ImageNet-100	-0.200	0.333	-0.067	**0.467**	0.067	**0.467**
ResNet-50(Pretrained)	ImageNet-100	0.293	0.008	0.092	0.410	0.233	**0.883**
ResNet-50(Pretrained)	ImageNet-1k	0.109	-0.059	0.109	0.243	0.233	**0.617**
ViT(Pretrained)	ImageNet-100	-0.033	0.253	-0.055	0.626	0.516	**0.802**
ViT(Pretrained)	ImageNet-1k	0.030	0.364	-0.061	0.152	0.576	**0.848**
Average		-0.135	0.128	0.119	0.252	0.119	**0.661**

A short note on the recent theoretical bounds

Same-class sampling has been also utilized in recent theoretical works where theoretical bounds are derived to connect contrastive learning and supervised learning (Arora et al. 2019; Nozawa and Sato 2021; Ash et al. 2021; Bao, Nagano, and Nozawa 2022). Unlike the practical and popular $\mathcal{T}_{\text{aug}}$, the supervised $\mathcal{T}_{\text{class}}$ provides strong structures and enables the deriving of meaningful results. All of the theoretical bounds, however, fail to correlate well with the downstream-task performance (see Table 5.4). (Also, ash2021investigating wrote "When using class information for sampling positives, however, the performance trends are somewhat unexpected." in Section 4 Experiments (Vision experiments, Figure 4).) Furthermore, contrastive training based on $\mathcal{T}_{\text{class}}$ does not guarantee a high performance as we will discuss in Section 5.4. Overall, the theoretical works are insightful, but somewhat disconnected from the practical issue of downstream-task performance.

Table 5.4. Theoretical upper bounds of the supervised loss for CDP dataset. All the bounds are determined based on the same variables, including the batch size, the number of class, and the contrastive loss. Since we fix the batch size and the number of classes, only the contrastive loss affects the bounds. Thus, all bounds have the same correlation coefficient of $\rho = -0.409$ and $\tau_K = -0.182$. We follow the official implementation codes of (Bao, Nagano, and Nozawa 2022).

Model	Training loss	Temperature	Acc. (%)	(Arora et al. 2019)	(Nozawa and Sato 2021)	(Ash et al. 2021)	(Bao, Nagano, and Nozawa 2022)
ResNet-18	$\mathcal{L}_{\text{SimCLR}}$	0.1	42.64	-399.448	1.931	-911.233	0.807
ResNet-18	$\mathcal{L}_{\text{SimCLR}}$	0.2	46.27	4830.149	5.277	11018.672	2.447
ResNet-18	$\mathcal{L}_{\text{SimCLR}}$	0.3	49.90	7315.527	6.867	16688.383	3.226
ResNet-50	$\mathcal{L}_{\text{SimCLR}}$	0.1	44.45	213.227	2.323	486.419	0.999
ResNet-50	$\mathcal{L}_{\text{SimCLR}}$	0.2	50.01	5250.332	5.546	11977.204	2.579
ResNet-50	$\mathcal{L}_{\text{SimCLR}}$	0.3	46.80	7553.165	7.019	17230.492	3.301
ResNet-18	$\mathcal{L}_{\text{class}}$	0.1	99.15	-159.413	2.084	-363.658	0.883
ResNet-18	$\mathcal{L}_{\text{class}}$	0.2	99.26	1089.677	2.883	2485.801	1.274
ResNet-18	$\mathcal{L}_{\text{class}}$	0.3	99.13	3778.281	4.604	8619.119	2.117
ResNet-50	$\mathcal{L}_{\text{class}}$	0.1	98.60	-167.034	2.079	-381.042	0.880
ResNet-50	$\mathcal{L}_{\text{class}}$	0.2	97.21	1133.557	2.911	2585.902	1.288
ResNet-50	$\mathcal{L}_{\text{class}}$	0.3	93.04	4061.839	4.785	9265.980	2.206

5.3.3. Minimizing task-irrelevant information (InfoMin) is not always necessary.

Existing belief 3:

For designing optimal views, task-irrelevant information needs to be discarded for a better generalization (Tian et al. 2020; Tsai et al. 2020; Xiao et al. 2020; Chen, Luo, and Li 2021).

Correction 3:

Task-irrelevant information does not necessarily harm the generalization of the downstream task.

The choice of augmentation is known to determine which type of invariance will be learned during contrastive learning (Tian et al. 2020; Tsai et al. 2020; Xiao et al. 2020; Chen, Luo, and Li 2021). tian2020makes formalized this idea into the InfoMin principle: 'a good set of views are those that share the minimal information necessary to perform well at the downstream task'. As shown in Figure 5.11, InfoMin claims that the choice of augmentation is critical for improving the downstream-task performance and the optimal augmentation strategy can be found by evaluating the mutual information. There are three stages for the shared information between two views, i.e., $I(X;Y)$: (1) missing information which leads to degraded performance due to $I(X;Y) < I(S;C)$; (2) excess noise which worsens generalization due to additional noise $I(X;Y) > I(S;C)$; (3) the sweet spot where the only information shared between two views is task-relevant and such information is complete, i.e. $I(X;Y) = I(S;C)$.

As an example of CDP dataset, we can describe the InfoMin principle as follows. We consider the case when the downstream-task is color (C_{color}). As shown in Figure 5.12(a), if we use the digit or position information for same-class sampling, the color label should be independently chosen for the two views. So, the task-specific information should be missing. On the contrary, as shown in Figure 5.12(c), if we use all

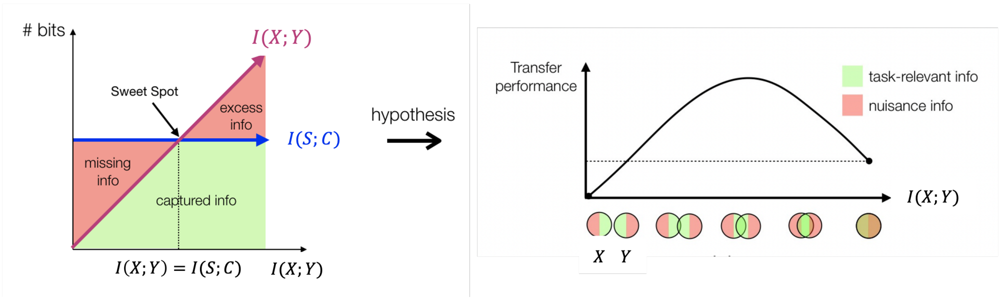


Figure 5.11. A description of InfoMin principle. The figure is adapted from Figure 1 of (Tian et al. 2020).

the information for same-class sampling in addition to the necessary color information, the representations should learn the excessive information for the downstream task of color. Finally, the optimal performance should be achieved when we utilize the positive pairing method corresponding to the sweet spot as shown in Figure 5.12(b).

Same-class Sampling for Positive Pairing $\mathcal{T}_{\text{class}}$

In the first experiment, we have investigated the CDP dataset where $\mathcal{T}_{\text{class}}^{\text{color}}$, $\mathcal{T}_{\text{class}}^{\text{digit}}$, $\mathcal{T}_{\text{class}}^{\text{position}}$, and $\mathcal{T}_{\text{class}}^{\text{all}}$ are considered for training and C_{color}, C_{digit}, C_{position}, and C_{all} are considered as the downstream task. The results are shown in Figure 5.14. As an example for ResNet-18, it can be seen that when $\mathcal{T}_{\text{class}}^{\text{digit}}$ is used for training, the performance for classifying color is 80.6%. If the InfoMin holds strongly, we would expect only the diagonal elements (same information for training and evaluation) to achieve a high performance. But the result shows that there are many non-diagonal elements that achieve a high performance. For instance, we can see that the performance of C_{digit} is higher when $\mathcal{T}_{\text{class}}^{\text{all}}$ is used for training (99.2%; four types of information are retained in the representation) than when $\mathcal{T}_{\text{class}}^{\text{digit}}$ is used for training (98.9%). Post-training MI estimation results are provided in Figure 5.15.

There is another interesting topic that can be noticed from Figure 5.14. When a specific positive pairing is used for training (e.g. $\mathcal{T}_{\text{class}}^{\text{color}}$), we would expect only the corresponding information (e.g. C_{color}) to be learned in the representation. The results in Figure 5.14, however, show that task-irrelevant information is frequently learned in the representation regardless of the positive pairing chosen for training. In particular, position information is always learned in our example. This indicates that targeting only for a specific type of information in contrastive learning might be quite challenging.

We conclude that the InfoMin principle is not always necessarily based on empirical results. However, it does not completely ignore the potential of mutual information for choosing the better data augmentation method because our results can be dependent on the particular choice of dataset. By utilizing the CDP dataset, we can clearly control the shared information between two views during training and evaluate the learned information by linear evaluation for different target variables. However, the CDP dataset is somewhat simple compared to the practical classification datasets. So, the deep network might memorize all the information included in the inputs even though the information is not shared between two views. Thus, for a complete understanding of our results, we need more empirical investigation and theoretical discussion. We defer them as future works.

Augmentation-based Positive Pairing $\mathcal{T}_{\text{aug}}$

As the second experiment, we expand our experiment to two well-known augmentations of $\mathcal{T}_{\text{aug}}$. Following (Tian et al. 2020), we utilize color jittering and random resized crop augmentations by varying the strength parameter. The results are provided in Figure 5.16 and Figure 5.17. Considering that color jittering is not related to digit task nor position task, the results in Figure 5.16 indicate that the peak in the middle might not be relevant to InfoMin. Similar results can be found for random resized crop. Considering that random resized crop might be less relevant to the color task than to the digit task or position task, the results in Figure 5.17 indicate that the peak in the middle might not be relevant to InfoMin either. Based on our results, aligning the positive pairing method $\mathcal{T}$ and the downstream task C is not possible, and also it might not be always helpful. Finally, we achieve the same conclusions as in the first experiment can be arrived.

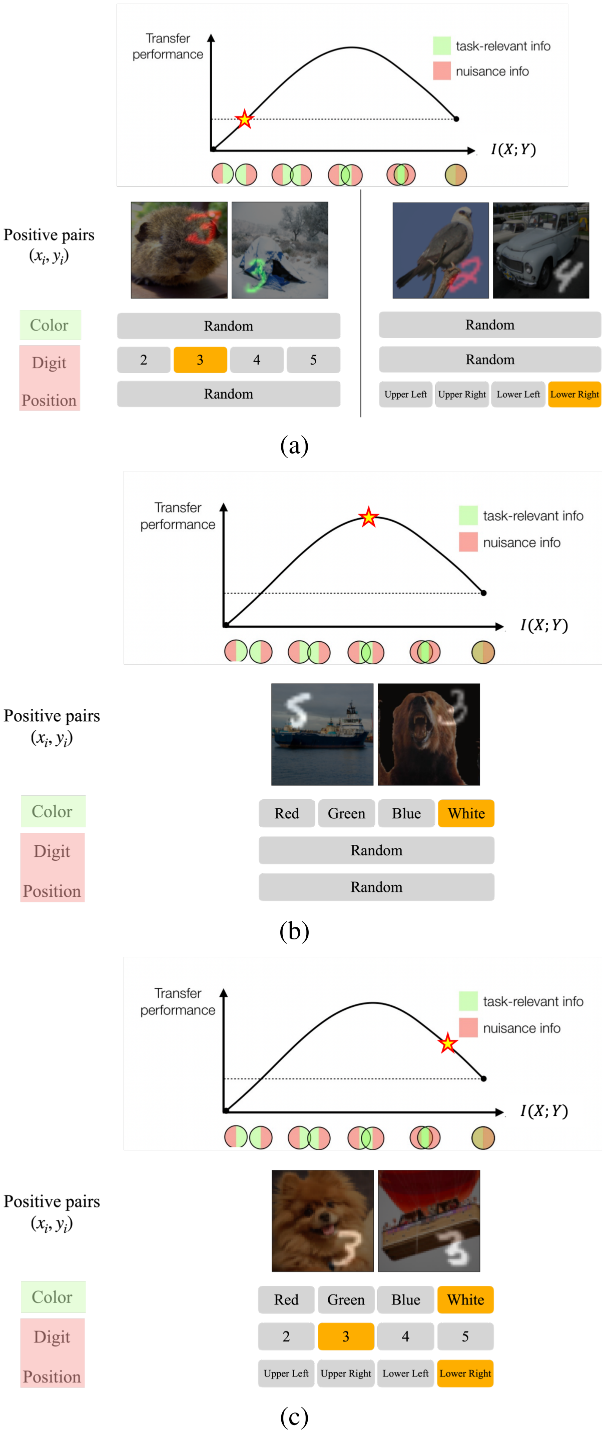


Figure 5.12. An example of InfoMin principle with CDP dataset.

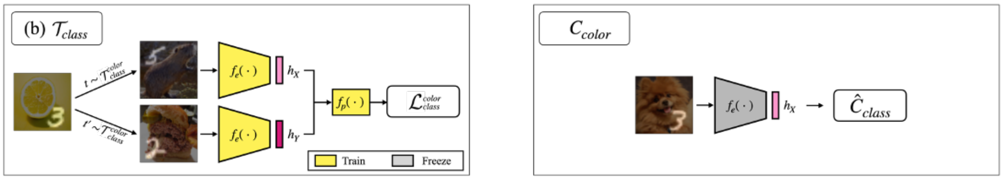

(b) $\mathcal{T}_{class}$
$t \sim \mathcal{T}^{color}_{class}$
$t' \sim \mathcal{T}^{color}_{class}$
$f_e(\cdot)$
h_X
h_Y
$f_p(\cdot)$
$\mathcal{L}^{color}_{class}$
Train
Freeze
C_{color}
$f_e(\cdot)$
h_X
$\hat{C}_{class}$

Figure 5.13. A summary of experimental setups. We first train the encoder network based on same-calss sampling for positive pairing. Then, we evaluate the learned representations for individual classification tasks.

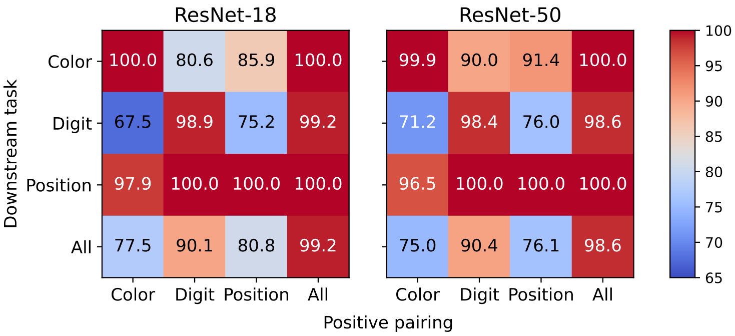


Figure 5.14. Linear evaluation performance of CDP dataset for task-dependent training. The task in x-axis indicates the positive pairing $\mathcal{T}$ used for training. We choose one of $\left\{\mathcal{T}^{\text{color}}_{\text{class}}, \mathcal{T}^{\text{digit}}_{\text{class}}, \mathcal{T}^{\text{position}}_{\text{class}}, \mathcal{T}^{\text{all}}_{\text{class}}\right\}$. The task in y-axis indicates the evaluated downstream task C. We choose one of $\{C_{\text{color}}, C_{\text{digit}}, C_{\text{position}}, C_{\text{all}}\}$.

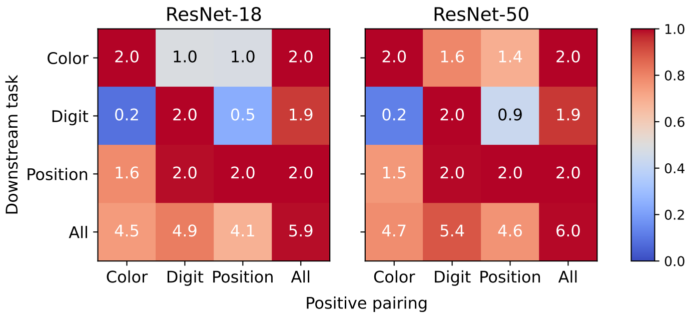


Figure 5.15. Post-training MI estimation results for the experiment cases in Figure 5.14.

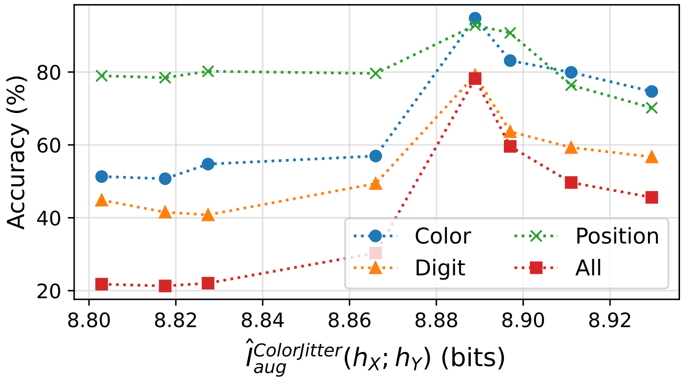


Figure 5.16. Linear evaluation performance when $\mathcal{T}_{\text{aug}}^{\text{ColorJitter}}$ is used for training. We have tuned the strength of color jittering in a way similar to (Tian et al. 2020). While we have found a similar result for the downstream task of color classification, the peak in the middle was found also for the other three tasks. Considering that color jittering is not related to digit task or position task, our result indicates that the peak in the middle might not be relevant to InfoMin.

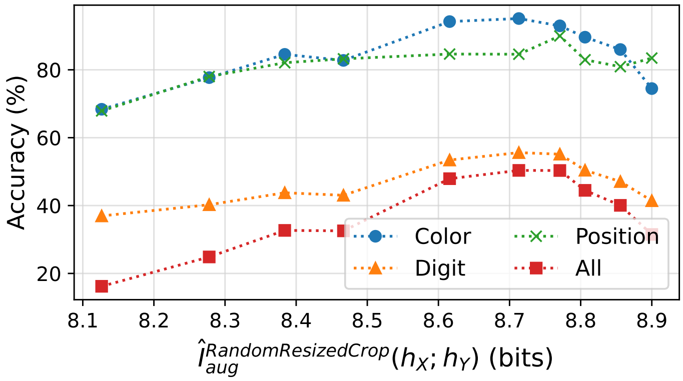


Figure 5.17. Linear evaluation performance when $\mathcal{T}_{\text{aug}}^{\text{RandomResizedCrop}}$ is used for training. We have tuned the minimum scale parameter of random resized crop in a way similar to (Tian et al. 2020).

5.4. Discussion

It has been common to analyze contrastive learning with MI, where MI is estimated for the $p_{\text{aug}}(x, y)$ that corresponds to the unsupervised positive pairing $\mathcal{T}_{\text{aug}}$ (the augmentation scheme applied during the training) (Oord, Li, and Vinyals 2018; Tschannen et al. 2019; Bachman, Hjelm, and Buchwalter 2019; Tian, Krishnan, and Isola 2020; Tian et al. 2020; Tsai et al. 2020). As shown in Section 5.3.2, however, there is no obvious reason for the commonly used MI to have a strong and consistent relationship with the downstream task performance. It will be more prudent to perform an analysis based on the post-training MI, $\hat{I}_{\text{class}}(h_X; h_Y)$. Because $\hat{I}_{\text{class}}(h_X; h_Y)$ is associated with $p_{\text{class}}(x, y)$ that is dependent on the downstream task's class information only, it is an adequate metric for investigating the factors that can affect the downstream task performance.

Downstream task's MI is an excellent performance metric, but it is not an effective learning objective.

Because we have observed in Section 5.3.2 that $\hat{I}_{\text{class}}(h_X; h_Y)$ is the most effective metric for downstream task's linear evaluation performance, it is reasonable to ask if the corresponding loss $\mathcal{L}_{\text{class}}$ in Figure 5.1(b) can learn a superior representation and achieve a better performance. A quick answer for this question is 'no'. Our experimental results are summarized in Table 5.5. Surprisingly, a carefully designed unsupervised learning can outperform a supervised contrastive learning that is based on the downstream-task information only. Here, a careful design basically means a well-crafted augmentation where the augmentation may have been designed in a heuristic manner or through an extensive tuning. We can see that the supervised loss $\mathcal{L}_{\text{class}}$ is outperformed by carefully designed unsupervised losses for two out of three cases. Even though $\hat{I}_{\text{class}}(h_X; h_Y)$ is a superior performance metric, the corresponding $\mathcal{L}_{\text{class}}$ is not necessarily a superior loss for learning representation. Furthermore, it is surprising to note that $\mathcal{L}_{\text{class}}$ is a supervised loss while the compared losses are unsupervised losses. Despite using the exact task information for the training, $\mathcal{L}_{\text{class}}$ performs worse than the carefully designed unsupervised learning methods.

A possible explanation can be related to the fact that $\mathcal{L}_{\text{class}}$ utilizes the minimum amount of information that is related to the task. While a high performing network must have its representation express the downstream-task information very well as we have shown in Section 5.3.2, the *training* of such a network requires additional learning signals on top of the basic downstream-task information. This explanation is also supported by the well known supervised loss $\mathcal{L}_{\text{SupCon}}$ that is proposed in (Khosla et al. 2020). Even though not shown in Table 5.5, the popular supervised loss $\mathcal{L}_{\text{SupCon}}$ easily outperforms the $\mathcal{L}_{\text{class}}$. In general, $\mathcal{L}_{\text{SupCon}}$ outperforms the unsupervised losses as well. $\mathcal{L}_{\text{SupCon}}$ is a supervised loss just like $\mathcal{L}_{\text{class}}$, but it experiments with known unsupervised augmentations and choose the high-performing augmentations to be used in addition to the class information.

Overall, we can conclude the followings for learning representation. (1) Using downstream-task information only (supervised) can be outperformed by a careful use of well-designed learning signals (unsupervised). (2) When supervised learning is allowed, both downstream-task information (i.e., class label) and well-designed learning signals (e.g., high-performance augmentations) should be used together to achieve the best performance.

Additionally, we would like to make it clear how our result is different from the work of (Tschannen et al. 2019). It has been already pointed out by (Tschannen et al. 2019) that MI alone might not be sufficient for learning effective representations for downstream tasks. The analysis method in the work, however, was not rigorous in that only a particular choice of augmentation and the corresponding joint distribution $p_{\text{aug}}(x, y)$ were studied. Without addressing exactly what information is shared by $p_{\text{aug}}(x, y)$, the analysis can be quite misleading. Furthermore, only $\mathcal{L}_{\text{aug}}$ was considered as the training objective in the work. As we have shown in Section 5.3.2, any analysis based on $\mathcal{L}_{\text{aug}}$ can be misleading because the information corresponding to the $p_{\text{aug}}(x, y)$ might not be sufficiently relevant to the downstream-task information anyway. In our work, we have considered $\mathcal{L}_{\text{class}}$ that is definitely related to the desired downstream-task information. While we also conclude that MI is not sufficient for a successful representation learning, our result is different and broadens the results in (Tschannen et al. 2019) because we have developed and applied rigorous methods for analyzing MI in contrastive learning.

Table 5.5. Comparison of linear evaluation performance for a set of loss functions. Performance with $*$ indicates values reported in the existing works. Despite the superiority of $\mathcal{L}_{\text{class}}$ as a metric, generally it does not outperform the best known unsupervised losses.

Loss	$\mathcal{L}_{\text{class}}$	$\mathcal{L}_{\text{SimCLR}}$	$\mathcal{L}_{\text{aug,best-known}}$
CIFAR-10	93.1	93.0	**94.1*** (SWD (Chen, Luo, and Li 2021))
ImageNet-100	**87.4**	77.8	84.5* (MoCo-v2+MoCHi (Kalantidis et al. 2020))
ImagNet-1k	75.2	69.1* (Chen et al. 2020b)	**76.4*** (HCA (Xu et al. 2020))

Negative sampling for effective contrastive learning does not need to follow the marginal distribution

Recently, the limitations of MI-based contrastive learning have been becoming clear. Many of the recent works have developed non-contrastive learning methods that can outperform MI-based contrastive learning (Caron et al. 2020; Grill et al. 2020b; Zbontar et al. 2021). Even for contrastive learning, small modifications in the loss function have been shown to be useful (Yeh et al. 2021), indicating that the loss function's deviation from an exact MI formulation can be advantageous. Here, we additionally show that the viewpoint of *Noise Contrastive Estimation* (NCE) in (Gutmann and Hyvärinen 2010) can be more relevant for enhancing the performance of unsupervised representation learning than the viewpoint of InfoNCE.

For the contrastive learning to be equivalent to an MI maximization, the negative term (the denominator in Eq. (2.2)) normalized by $(2K-1)$ needs to be an asymptotic estimation of the partition function $Z(y)(=\mathbb{E}_{p(y)}[e^{f(x,y)}])$ (Poole et al. 2019). This requirement can be fulfilled by drawing the negative samples with a uniform distribution over the entire training dataset. In practice, the negative samples in Eq. (2.2) are chosen as the samples in the mini-batch, primarily for the computational efficiency.

In contrast to the viewpoint of MI maximization, the viewpoint of *Noise Contrastive Estimation* (NCE) in (Gutmann and Hyvärinen 2010) does not require the negative samples to be drawn from the marginal distribution. Instead, the negative samples can be drawn from any reasonable distribution including random noise such as Gaussian noise. Interestingly, both viewpoints were addressed in the original CPC work (Oord, Li, and Vinyals 2018), but the relationship between the two viewpoints was not clarified. Here, we provide an experiment to show that the negative samples do not need to be drawn from the marginal distribution. In fact, we can enhance the performance of contrastive learning by carefully manipulating the negative sampling.

Before proceeding, we define four new datasets. CIFAR-5A and CIFAR-5B are disjoint datasets created from CIFAR-10. CIFAR-5A contains all the examples of the first five classes of CIFAR-10 and CIFAR-5B contains all the examples of the last five classes of CIFAR-10. CIFAR-50A and CIFAR-50B are created in a similar way from CIFAR-100 (first fifty classes of CIFAR-100 and last fifty classes of CIFAR-100).

The experimental results are shown in Table 5.6. The positive pairs are always drawn from the original dataset $\mathcal{D}$ (CIFAR-5A or CIFAR-50A), but the negative samples are drawn from the negative sampling dataset $\mathcal{D}^-$. As expected, performance degradation can be observed when $\mathcal{D}^-$ is one of PACS-(cartoon, art, photo, and sketch) (Li et al. 2017) or uniform random noise. When $\mathcal{D}^-$ is CIFAR-5B, however, the performance is improved by 1.92$\%$. The same observations can be made for CIFAR-50A, with the improvement of 0.77$\%$. The experiment results indicate that we can improve the linear evaluation performance by carefully choosing $\mathcal{D}^-$ for negative sampling. In our

Table 5.6. The effect of negative sampling dataset $\mathcal{D}^-$. Linear evaluation performance can be affected by choosing negative samples from a related or an unrelated dataset. (a) CIFAR-5A: For contrastive learning of CIFAR-5A dataset, the best performance is achieved by choosing the negative samples from CIFAR-5B dataset (i.e., not from CIFAR-5A dataset). (b) CIFAR-50A: For contrastive learning of CIFAR-50A dataset, the best performance is achieved by choosing the negative samples from CIFAR-50B dataset (i.e., not from CIFAR-50A dataset).

(a) $\mathcal{D}$ = CIFAR-5A

$\mathcal{D}^-$	CIFAR-5A (Baseline: InfoNCE loss)	CIFAR-5B	PACS-C	PACS-A	PACS-P	PACS-S	Uniform random
Accuracy (%)	85.70	**87.62**	83.14	81.98	81.14	80.86	79.80

(b) $\mathcal{D}$ = CIFAR-50A

$\mathcal{D}^-$	CIFAR-50A (Baseline: InfoNCE loss)	CIFAR-50B	PACS-C	PACS-A	PACS-P	PACS-S	Uniform random
Accuracy (%)	59.56	**60.34**	49.52	51.16	50.44	43.40	33.92

experiments, the performance was enhanced by choosing a dataset whose distribution slightly diverges from the true marginal distribution (CIFAR-5B and CIFAR-50B are not the marginals but at least they come from the same source datasets of CIFAR-10 and CIFAR-100).

While a high performing network must have its representation express the downstream-task information very well as we have shown in Section 5.3.2, the *training* of such a network requires additional learning signals regardless of the presence of the downstream-task information. All the cases discussed above strengthen the idea that MI of downstream task is an outstanding metric but clearly not an excellent learning objective.

Combining $\mathcal{L}_{\text{aug}}$ and $\mathcal{L}_{\text{class}}$

We observed that MI of downstream tasks is an outstanding metric but clearly not an excellent learning objective. Then, we could ask if there should be an improved strategy by combining two metrics. A similar approach has already been investigated in (Khosla et al. 2020), called SupCon. SupCon optimizes the InfoNCE loss when the positive pairs are defined based on the supervised labels. This loss is exactly what we used as $\mathcal{L}_{\text{class}}$. However, they have used additional augmentation methods. Based on the results of SupCon, we expect that we can improve the downstream-task performance by combining $\mathcal{L}_{\text{aug}}$ and $\mathcal{L}_{\text{class}}$.

Rethinking contrastive learning - is it really an unsupervised learning method?

If the only metric that is truly effective for predicting downstream task's performance is the downstream-task information itself as we have shown in Section 5.3.2, how is it possible to learn effective representations in an unsupervised way? First of all, it is crucial to recognize that the augmentation design is not completely unsupervised because the validation performance (linear evaluation performance) is used for the selection of augmentation design. The validation data does not directly affect the network parameters (i.e., no gradient descent with the validation data), but it indirectly affects the network parameters because the selection of augmentation design affects the joint distribution $p(x, y)$, in turn $p(x, y)$ defines the MI of the learning, and the MI affects the goal of learning as well as the learning dynamics.

The success of contrastive learning methods, and the closely related non-contrastive learning methods, seem to be due to two main reasons. First, compared to the early techniques such as pretext learning (Doersch, Gupta, and Efros 2015; Pathak et al. 2016; Noroozi and Favaro 2016; Gidaris, Singh, and Komodakis 2018), augmentation design can be successfully and efficiently completed within a limited design search space. Typically, effective augmentation techniques for supervised learning are already known for each application area, and properly combining the known techniques is a good start for achieving a high performance with an unsupervised contrastive learning. Second, the learned representation seems to generalize better than the traditional methods. This seems to be surprisingly true for the popular benchmark problems, but a careful study is still needed to confirm it for a wider set of applications and datasets.

Despite the amazing success of contrastive learning, it still remains open to develop a further advanced representation learning framework where a heuristic search of augmentation design per application area can be avoided.

5.5. Conclusion

In this work, we have examined three existing beliefs on mutual information in contrastive learning. For a rigorous investigation, we made use of same-class sampling, post-training MI estimation, and CDP dataset. We have empirically shown that the three existing beliefs are incorrect or misleading, and provided adequate corrections. We discussed how maximizing the MI of downstream task's information is necessary but not sufficient for an unsupervised representation learning. A limitation of our study is that we have focused on image classification as the only downstream task. Our study can be extended to other downstream tasks such as object detection and to other datasets such as NLP datasets.

Chapter 6. Conclusion

In this dissertation, we have examined the role of mutual information in contrastive representation learning by developing a rigorous investigation method. To access the true MI values for real-world datasets, such as image and text datasets whose probability distribution is intractable, we suggested a same-class sampling for positive pairing. Based on the same-class sampling, we first evaluated the accuracy of the variational MI estimators under various scenarios. As a result, we found that variational MI estimators do not provide the same behavior for images and texts compared to the toy dataset drawn from the multivariate Gaussian distribution. Thus, it is necessary to have access to the true MI value when we use MI as the metric for the purpose of analysis. Finally, we examined three existing beliefs on MI in contrastive learning. To make use of the exact true MI values for the image datasets, we proposed the CDP dataset satisfying the assumption for the equality condition for same-class sampling. In addition, we define the post-training MI estimation phase to prevent the effect of the encoder network on the estimated MI values. Based on the analysis framework, we found that (1) a small batch size limits the training loss, but it limits neither the information in the learned representation nor the downstream-task performance, (2) the only metric that is strongly relevant to the downstream-task performance is the MI of the downstream-task information itself, and (3) task-irrelevant information does not necessarily harm the generalization of the downstream task. Overall, we found that there are a few fundamental problems that need to be addressed for the current framework of unsupervised contrastive learning. Any change in augmentation design affects the joint distribution $p(x, y)$. In turn, the shared information between X and Y (i.e., MI value) is affected, and eventually the learning is affected. Because the whole process is based on validation performance, where the augmentation design with a high downstream task performance is manually chosen as in any hand-tuning, there are issues that need to be carefully addressed.

In summary, our results claim that MI of downstream-task is an outstanding metric but clearly not an excellent learning objective. Even, the learning objective does not need to follow the MI formulation at all. It highlights the previous approaches that interpret or design contrastive learning from the perspective of maximizing MI are highly misleading. Certainly, contrastive loss follows the formulation of the InfoNCE estimator. However, it only provides the estimation of mutual information between the learned representations. Thus, we need an elaborate analysis framework to understand how MI is related to contrastive learning, instead of simply evaluating the estimated values. When we have the estimated values without the true values, we cannot make any scientific conclusion because the estimated values can be erroneous. Even though we have a theoretical bound for the generalization gap between the estimation and ground-truth values, DNN-based methods are known as not satisfying the bounds because of the complex optimization process. Despite these issues, previous studies have focused on the mathematical equivalence of the contrastive loss and InfoNCE estimator only, and they overlooked carefully examining what role mutual information plays in the success of contrastive learning.

Our study has been motivated by this fundamental question — "Is minimizing contrastive loss equivalent to maximizing MI? Can we attribute the success of contrastive learning to maximizing MI?". Even though we found no clear answer to this question, mutual information has been referred to as the key factor for improving contrastive representation learning. Based on the carefully designed analysis framework, we showed the previous beliefs related to the role of mutual information in contrastive learning are highly misleading and a rigorous investigation must be conducted before setting the assumptions. Our results suggest mutual information does not contribute to the success of contrastive learning, and the loss function or augmentation design does not need to follow the formulation of MI. Finally, we expect unsupervised learning could result a much better representations when we are not stuck in the frame of mutual information.

6.1. Limitations

In Chapter 4, we found that true MI and MI estimation are not available when the dataset includes too much nuisance. As such, MI would not be a proper choice when we want to analyze the dataset itself. Variational MI estimators only provide meaningful results when we have a well-trained encoder $g(\cdot)$.

In Chapter 5, we focused on image classification as the downstream task. Our analysis framework can be extended for investigating other downstream tasks, such as object detection or other datasets in other application fields.

We examined some particular existing beliefs related to MI and contrastive learning in Chapter 5. Even though we have considered the major topics, some items have not been discussed in this study. For example, (Chen, Luo, and Li 2021) observed the feature suppression effect.

Throughout this study, we largely depend on the specific choice of positive pairing, called same-class sampling. While same-class sampling allows us to make use of the true MI values for any datasets under a mild assumption, it requires the label information, and we cannot use this method when we have no proper discretized labels. In addition, same-class sampling provides the exact MI value only if the dataset satisfies the assumption. In this study, we avoid these limitations by adopting image classification for the downstream task and defining a synthetic dataset that has an error-free classification function.

6.2. Future works

In our study, we have raised a counterargument for the common beliefs on mutual information in contrastive representation learning. In conclusion, we believe that MI-motivated design principles might not work well for improving the representation learning itself, but MI could be a good metric for analyzing the deep representations. As a future work, we plan to use MI to explain the effectiveness of deep networks, such as residual connection and pre-training.

Bibliography

Achille, A., and Soatto, S. 2018. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research* 19(1):1947–1980.

Arora, S.; Khandeparkar, H.; Khodak, M.; Plevrakis, O.; and Saunshi, N. 2019. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*.

Ash, J. T.; Goel, S.; Krishnamurthy, A.; and Misra, D. 2021. Investigating the role of negatives in contrastive representation learning. *arXiv preprint arXiv:2106.09943*.

Bachman, P.; Hjelm, R. D.; and Buchwalter, W. 2019. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*.

Bao, H.; Nagano, Y.; and Nozawa, K. 2022. On the surrogate gap between contrastive and supervised losses. In *International Conference on Machine Learning*, 1585–1606. PMLR.

Barber, D., and Agakov, F. 2003. Information maximization in noisy channels: A variational approach. *Advances in Neural Information Processing Systems* 16.

Bardes, A.; Ponce, J.; and LeCun, Y. 2021. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*.

Belghazi, M. I. h.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, R. D. 2018. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*.

Bell, A. J., and Sejnowski, T. J. 1995. An information-maximization approach to blind separation and blind deconvolution. *Neural computation* 7(6):1129–1159.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9650–9660.

Chen, X., and He, K. 2020. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*.

Chen, X., and He, K. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15750–15758.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; and Abbeel, P. 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*.

Chen, T. Q.; Li, X.; Grosse, R. B.; and Duvenaud, D. K. 2018. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2610–2620.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020c. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Chen, T.; Luo, C.; and Li, L. 2021. Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems* 34.

Chen, X.; Xie, S.; and He, K. 2021. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9640–9649.

Cheng, P.; Hao, W.; Dai, S.; Liu, J.; Gan, Z.; and Carin, L. 2020. Club: A contrastive log-ratio upper bound of mutual information. *arXiv preprint arXiv:2006.12013*.

Cover, T. M. 1999. *Elements of information theory*. John Wiley & Sons.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 702–703.

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine* 29(6):141–142.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, 1422–1430.

Donsker, M. D., and Varadhan, S. S. 1983. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics* 36(2):183–212.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N.

2021. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

Fraser, A. M., and Swinney, H. L. 1986. Independent coordinates for strange attractors from mutual information. *Physical review A* 33(2):1134.

Gao, T.; Yao, X.; and Chen, D. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; and He, K. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.

Goyal, P.; Duval, Q.; Reizenstein, J.; Leavitt, M.; Xu, M.; Lefaudeux, B.; Singh, M.; Reis, V.; Caron, M.; Bojanowski, P.; Joulin, A.; and Misra, I. 2021. Vissl. `https://github.com/facebookresearch/vissl`.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020a. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P. H.; Buchatskaya, E.; Doersch, C.; Pires, B. A.; Guo, Z. D.; Azar, M. G.; et al. 2020b. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*.

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

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Henaff, O. 2020. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, 4182–4192. PMLR.

Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; and Oh, S. J. 2021. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11936–11945.

Hermann, K., and Lampinen, A. 2020. What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems* 33:9995–10006.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Hyvärinen, A., and Oja, E. 2000. Independent component analysis: algorithms and applications. *Neural networks* 13(4-5):411–430.

Jacobsen, J.-H.; Smeulders, A.; and Oyallon, E. 2018. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*.

Kalantidis, Y.; Sariyildiz, M. B.; Pion, N.; Weinzaepfel, P.; and Larlus, D. 2020. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems* 33:21798–21809.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems* 33:18661–18673.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kornblith, S.; Chen, T.; Lee, H.; and Norouzi, M. 2021. Why do better loss functions lead to less transferable features? *Advances in Neural Information Processing Systems* 34:28648–28662.

Kraskov, A.; Stögbauer, H.; and Grassberger, P. 2004. Estimating mutual information. *Physical review E* 69(6):066138.

Li, D.; Yang, Y.; Song, Y.-Z.; and Hospedales, T. M. 2017. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, 5542–5550.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022.

Loshchilov, I., and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Ma, Z., and Collins, M. 2018. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812*.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

McAllester, D., and Stratos, K. 2020. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, 875–884.

Misra, I., and Maaten, L. v. d. 2020. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6707–6717.

Mitrovic, J.; McWilliams, B.; Walker, J.; Buesing, L.; and Blundell, C. 2020. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*.

Moyer, D.; Gao, S.; Brekelmans, R.; Galstyan, A.; and Ver Steeg, G. 2018. Invariant representations without adversarial training. *Advances in Neural Information Processing Systems* 31.

Nguyen, T.; Raghu, M.; and Kornblith, S. 2020. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. *arXiv preprint arXiv:2010.15327*.

Nguyen, T.; Raghu, M.; and Kornblith, S. 2022. On the origins of the block structure phenomenon in neural network representations. *arXiv preprint arXiv:2202.07184*.

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

Noroozi, M., and Favaro, P. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, 69–84. Springer.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, 271–279.

Nozawa, K., and Sato, I. 2021. Understanding negative samples in instance discriminative self-supervised representation learning. *Advances in Neural Information Processing Systems* 34.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

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

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

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2536–2544.

Poole, B.; Ozair, S.; Van Den Oord, A.; Alemi, A.; and Tucker, G. 2019. On variational bounds of mutual information. In *International Conference on Machine Learning*, 5171–5180.

Purushwalkam, S., and Gupta, A. 2020. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*.

Shwartz-Ziv, R., and Tishby, N. 2017. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*.

Song, J., and Ermon, S. 2019. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations*.

Song, J., and Ermon, S. 2020. Multi-label contrastive predictive coding. *arXiv preprint arXiv:2007.09852*.

Song, L.; Smola, A.; Gretton, A.; Bedo, J.; and Borgwardt, K. 2012. Feature selection via dependence maximization. *Journal of Machine Learning Research* 13(5).

Sordoni, A.; Dziri, N.; Schulz, H.; Gordon, G.; Bachman, P.; and Des Combes, R. T. 2021. Decomposed mutual information estimation for contrastive representation learning. In *International Conference on Machine Learning*, 9859–9869. PMLR.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, 776–794. Springer.

Tishby, N., and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, 1–5. IEEE.

Tomasev, N.; Bica, I.; McWilliams, B.; Buesing, L.; Pascanu, R.; Blundell, C.; and Mitrovic, J. 2022. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021a. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357. PMLR.

Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; and Jégou, H. 2021b. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 32–42.

Tsai, Y.-H. H.; Wu, Y.; Salakhutdinov, R.; and Morency, L.-P. 2020. Self-supervised learning from a multi-view perspective. *arXiv preprint arXiv:2006.05576*.

Tschannen, M.; Djolonga, J.; Rubenstein, P. K.; Gelly, S.; and Lucic, M. 2019. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*.

Wang, T., and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 9929–9939. PMLR.

Wang, F., and Liu, H. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2495–2504.

Wang, B.; Wang, S.; Cheng, Y.; Gan, Z.; Jia, R.; Li, B.; and Liu, J. 2020. Infobert: Improving robustness of language models from an information theoretic perspective. *arXiv preprint arXiv:2010.02329*.

Wightman, R. 2019. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`.

Wu, Z.; Xiong, Y.; Yu, S.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance-level discrimination. *arXiv preprint arXiv:1805.01978*.

Wu, M.; Mosse, M.; Zhuang, C.; Yamins, D.; and Goodman, N. 2020a. Conditional negative sampling for contrastive learning of visual representations. *arXiv preprint arXiv:2010.02037*.

Wu, M.; Zhuang, C.; Mosse, M.; Yamins, D.; and Goodman, N. 2020b. On mutual information in contrastive learning for visual representations. *arXiv preprint arXiv:2005.13149*.

Xiao, T.; Wang, X.; Efros, A. A.; and Darrell, T. 2020. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659*.

Xie, E.; Ding, J.; Wang, W.; Zhan, X.; Xu, H.; Sun, P.; Li, Z.; and Luo, P. 2021. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8392–8401.

Xu, A., and Raginsky, M. 2017. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in Neural Information Processing Systems* 30.

Xu, Y.; Zhao, S.; Song, J.; Stewart, R.; and Ermon, S. 2019. A theory of usable information under computational constraints. In *International Conference on Learning Representations*.

Xu, H.; Zhang, X.; Li, H.; Xie, L.; Xiong, H.; and Tian, Q. 2020. Seed the views: Hierarchical semantic alignment for contrastive representation learning. *arXiv preprint arXiv:2012.02733*.

Yan, X.; Misra, I.; Gupta, A.; Ghadiyaram, D.; and Mahajan, D. 2020. Clusterfit:

Improving generalization of visual representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6509–6518.

Yeh, C.-H.; Hong, C.-Y.; Hsu, Y.-C.; Liu, T.-L.; Chen, Y.; and LeCun, Y. 2021. Decoupled contrastive learning. *arXiv preprint arXiv:2110.06848*.

Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; and Deny, S. 2021. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*.

Appendices

A. Implementation details of Chapter 5

A.1. SimCLR augmentation ($\mathcal{T}_{\text{SimCLR}}$)

As a representative case of unsupervised positive pairing $\mathcal{T}_{\text{aug}}$, we adopt the SimCLR augmentation (Chen et al. 2020b). The details of the code implementation of each dataset are provided here. We use PyTorch and torchvision library.

CDP dataset

```
img_size = 32; strength = 0.5
color_jitter = torchvision.transforms.ColorJitter(
    brightness=0.8 * strength, contrast=0.8 * strength,
    saturation=0.8 * strength, hue=0.2 * strength)
transform = torchvision.transforms.Compose([
    torchvision.transforms.RandomResizedCrop(size=img_size),
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.RandomApply([color_jitter], p=0.8),
    torchvision.transforms.RandomGrayscale(p=0.2),
    torchvision.transforms.ToTensor()])
```

For Table 5.5, We empirically found the $\mathcal{T}_{\text{aug}}$ shown below by searching for the performance.

```
transform = torchvision.transforms.Compose([
    torchvision.transforms.RandomResizedCrop(
        size=img_size, scale=(0.5, 1.0)),
    torchvision.transforms.RandomApply([color_jitter], p=0.5),
    torchvision.transforms.ToTensor()])
```

CIFAR-10

```
img_size = 32; strength = 0.5
color_jitter = torchvision.transforms.ColorJitter(
    brightness=0.8 * strength, contrast=0.8 * strength,
    saturation=0.8 * strength, hue=0.2 * strength)
transform_train = torchvision.transforms.Compose([
    torchvision.transforms.RandomResizedCrop(size=img_size),
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.RandomApply([color_jitter], p=0.8),
    torchvision.transforms.RandomGrayscale(p=0.2),
    torchvision.transforms.ToTensor(),
    torchvision.transforms.Normalize(
        mean=[0.4914, 0.4822, 0.4465],
        std=[0.2023, 0.1994, 0.2010])])
```

ImageNet

```
img_size = 224; strength = 1.; ksize = 23
color_jitter = torchvision.transforms.ColorJitter(
    brightness=0.8 * strength, contrast=0.8 * strength,
    saturation=0.8 * strength, hue=0.2 * strength)
transform = torchvision.transforms.Compose([
    torchvision.transforms.RandomResizedCrop(
        size=img_size, scale=(0.2, 1.0)),
    torchvision.transforms.RandomHorizontalFlip(),
    torchvision.transforms.RandomApply([color_jitter], p=0.8),
    torchvision.transforms.RandomGrayscale(p=0.2),
    GaussianBlur(kernel_size=ksize),
    torchvision.transforms.ToTensor(),
    torchvision.transforms.Normalize(
        mean=[0.485, 0.456, 0.406],
        std=[0.229, 0.224, 0.225])])
```

A.2. Full results of Table 5.3

In Table 5.3, the summary of seven experiments is provided. Here, we report the full results of the seven experiments. For alignment and uniformity, a smaller value is better (↓). For tolerance and linear CKA, a higher value is better (↑). Note that class label information is utilized by tolerance, linear CKA, and $\hat{I}_{\text{class}}(h_X; h_Y)$.

CDP, CIFAR-10, ImageNet-100 with three different temperatures

For CDP, CIFAR-10, and ImageNet-100, we train the encoders of ResNet-18/50 from scratch following the setups in Section 5.2.1. We test three temperature parameters for each dataset. The results are shown below.

Table A.1. Metrics evaluated on CDP dataset.

Model	Temperature	Acc. (%)	Alignment ↓	Uniformity ↓	Tolerance ↑	Linear CKA ↑	$\hat{I}_{\text{SimCLR}}(X;Y)$	$\hat{I}_{\text{class}}(X;Y)$
ResNet-18	0.1	42.64	0.196	-0.941	0.776	0.056	5.960	3.179
ResNet-18	0.2	46.27	0.227	-1.074	0.743	0.052	5.387	3.060
ResNet-18	0.3	49.90	0.218	-1.041	0.747	0.017	4.938	2.957
ResNet-50	0.1	44.45	0.184	-0.685	0.834	0.005	5.663	3.374
ResNet-50	0.2	50.01	0.226	-0.784	0.820	0.051	5.107	3.547
ResNet-50	0.3	46.80	0.214	-0.723	0.829	0.000	4.498	2.936
Pearson's ρ with Acc.			0.977	0.058	0.956	**0.992**	-0.988	0.990
Kendall's τ_K with Acc.			0.545	-0.061	0.485	0.333	-0.727	**0.545**

Table A.2. Metrics evaluated on CIFAR-10 dataset.

Model	Temperature	Acc. (%)	Alignment ↓	Uniformity ↓	Tolerance ↑	Linear CKA ↑	$\hat{I}_{\text{SimCLR}}(X;Y)$	$\hat{I}_{\text{class}}(X;Y)$
ResNet-18	0.1	90.03	0.380	-2.735	0.399	0.287	8.063	2.717
ResNet-18	0.3	91.11	0.449	-3.147	0.321	0.293	7.912	2.874
ResNet-18	0.5	90.97	0.427	-2.839	0.427	0.452	7.730	2.756
ResNet-50	0.1	92.06	0.403	-2.351	0.458	0.227	8.117	2.806
ResNet-50	0.3	92.97	0.562	-2.950	0.328	0.224	7.954	2.902
ResNet-50	0.5	93.01	0.467	-2.432	0.467	0.267	7.879	2.803
Pearson's ρ with Acc.			0.738	0.319	0.121	-0.503	-0.041	**0.634**
Kendall's τ_K with Acc.			0.600	0.067	0.333	-0.467	-0.067	**0.467**

Table A.3. Metrics evaluated on ImageNet-100 dataset.

Model	Temperature	Acc. (%)	Alignment ↓	Uniformity ↓	Tolerance ↑	Linear CKA ↑	$\hat{I}_{\text{SimCLR}}(X;Y)$	$\hat{I}_{\text{class}}(X;Y)$
ResNet-18	0.1	72.60	0.291	-2.265	0.497	0.329	8.364	3.394
ResNet-18	0.2	76.42	0.352	-2.593	0.438	0.375	8.375	3.907
ResNet-18	0.3	75.66	0.315	-2.268	0.515	0.405	8.313	3.857
ResNet-50	0.1	74.08	0.038	-0.270	0.941	0.272	8.412	3.967
ResNet-50	0.2	75.52	0.037	-0.277	0.943	0.332	8.347	4.186
ResNet-50	0.3	77.80	0.056	-0.408	0.914	0.338	8.403	4.263
Pearson's ρ with Acc.			-0.165	0.197	0.214	0.410	0.085	**0.805**
Kendall's τ_K with Acc.			0.200	-0.333	-0.067	**0.467**	0.067	**0.467**

Evaluations over pre-trained encoders: ImageNet-100 and ImageNet-1k

We additionally test a variety of pre-trained models loaded from (Goyal et al. 2021; Khosla et al. 2020; Wightman 2019). We inspect 16 pre-trained ResNet-50 models and 14 pre-trained ViT models. All models are pre-trained by ImageNet-1k dataset. We load the pre-trained models and evaluate the linear accuracy and the metrics. The results are shown below.

Table A.4. Metrics evaluated on ImageNet-100 dataset using pre-trained ResNet-50 models.

Algorithm	Acc. (%)	Alignment ↓	Uniformity ↓	Tolerance ↑	Linear CKA ↑	$\hat{I}_{\text{SimCLR}}(X;Y)$	$\hat{I}_{\text{class}}(X;Y)$
SupCon (Khosla et al. 2020)	94.40	0.107	-2.600	0.489	0.439	7.889	6.100
Supervised pretrained	93.00	0.701	-3.173	0.380	0.403	7.598	5.816
SwAV (Caron et al. 2020)	92.52	0.296	-1.659	0.636	0.282	8.544	5.560
DeepCluster-v2 (Caron et al. 2020)	92.38	0.244	-1.308	0.709	0.254	8.544	5.560
DINO (Caron et al. 2021)	92.22	0.433	-1.829	0.592	0.277	8.443	5.539
Barlow Twins (Zbontar et al. 2021)	90.80	0.477	-2.415	0.458	0.316	8.528	5.513
PIRL (Misra and Maaten 2020)	90.58	0.388	-3.387	0.361	0.452	8.584	5.480
SeLa-v2 (Caron et al. 2020)	89.50	0.208	-1.098	0.752	0.302	6.020	5.039
SimCLR (Chen et al. 2020b)	89.40	0.519	-3.032	0.336	0.425	8.669	5.546
MoCo-v2 (Chen et al. 2020c)	87.54	0.321	-2.820	0.497	0.413	8.592	5.490
NPID++ (Misra and Maaten 2020)	79.60	0.745	-2.637	0.423	0.303	8.190	4.792
MoCo (He et al. 2020)	76.94	0.701	-3.174	0.380	0.403	8.338	4.904
NPID (Wu et al. 2018)	76.68	0.745	-2.637	0.423	0.201	8.039	4.188
ClusterFit (Yan et al. 2020)	75.66	0.706	-3.019	0.321	0.199	8.016	4.155
RotNet (Gidaris, Singh, and Komodakis 2018)	66.90	0.625	-1.927	0.561	0.166	7.020	2.916
Jigsaw (Noroozi and Favaro 2016)	56.74	0.220	-0.486	0.888	0.076	6.339	2.510
Pearson's ρ with Acc.		-0.286	-0.265	-0.227	0.722	0.510	**0.967**
Kendall's τ_K with Acc.		-0.293	-0.008	0.092	0.410	0.233	**0.883**

Table A.5. Metrics evaluated on ImageNet-1k dataset using pre-trained ResNet-50 models.

Algorithm	Acc. (%)	Alignment ↓	Uniformity ↓	Tolerance ↑	Linear CKA ↑	$\hat{I}_{\text{SimCLR}}(X;Y)$	$\hat{I}_{\text{class}}(X;Y)$
SupCon (Khosla et al. 2020)	78.72	0.697	-2.560	0.479	0.302	8.722	7.783
Supervised pretrained	74.11	0.704	-3.169	0.369	0.373	8.378	6.761
SwAV (Caron et al. 2020)	74.78	0.298	-1.637	0.634	0.228	9.428	6.214
DeepCluster-v2 (Caron et al. 2020)	73.65	0.247	-1.284	0.708	0.177	9.416	6.232
DINO (Caron et al. 2021)	74.22	0.434	-1.802	0.590	0.225	9.313	6.133
Barlow Twins (Zbontar et al. 2021)	72.82	0.485	-2.394	0.454	0.240	9.407	6.157
PIRL (Misra and Maaten 2020)	70.51	0.400	-3.378	0.345	0.375	9.481	6.247
SeLa-v2 (Caron et al. 2020)	69.66	0.209	-1.064	0.756	0.218	7.354	5.774
SimCLR (Chen et al. 2020b)	69.12	0.536	-2.991	0.329	0.397	9.580	6.277
MoCo-v2 (Chen et al. 2020c)	63.89	0.333	-2.801	0.480	0.399	9.499	6.221
NPID++ (Misra and Maaten 2020)	56.60	0.845	-2.634	0.335	0.289	9.009	4.692
MoCo (He et al. 2020)	47.052	0.704	-3.169	0.369	0.373	9.155	4.907
NPID (Wu et al. 2018)	52.70	0.761	-2.634	0.417	0.192	8.821	3.836
ClusterFit (Yan et al. 2020)	48.81	0.710	-3.004	0.313	0.171	8.773	3.915
RotNet (Gidaris, Singh, and Komodakis 2018)	41.54	0.627	-1.913	0.553	0.143	7.696	2.802
Jigsaw (Noroozi and Favaro 2016)	30.85	0.221	-0.479	0.888	0.091	7.155	2.583
Pearson's ρ with Acc.		-0.175	-0.157	-0.132	0.451	0.535	**0.943**
Kendall's τ_K with Acc.		-0.109	0.059	0.109	0.243	0.233	**0.617**

Table A.6. Metrics evaluated on ImageNet-100 dataset using pre-trained ViT models.

Algorithm	Acc. (%)	Alignment ↓	Uniformity ↓	Tolerance ↑	Linear CKA ↑	$\hat{I}_{\text{SimCLR}}(X;Y)$	$\hat{I}_{\text{class}}(X;Y)$
Swin-B (Liu et al. 2021)	96.20	0.787	-3.663	0.502	0.559	8.073	6.222
Supervised pretrained (ViT-B/16) (Dosovitskiy et al. 2021)	95.36	0.565	-3.843	0.450	0.538	8.252	5.977
PiT-B (Heo et al. 2021)	94.62	0.880	-3.694	0.497	0.520	7.895	6.398
DeiT (ViT-B/16) (Touvron et al. 2021a)	94.30	0.833	-3.761	0.507	0.499	7.799	6.287
CaiT (XXS-36/16) (Touvron et al. 2021b)	93.90	0.644	-3.745	0.566	0.414	7.492	5.795
PiT-S (Heo et al. 2021)	93.76	0.820	-3.763	0.491	0.448	7.664	6.151
DeiT (ViT-S/16) (Touvron et al. 2021a)	93.42	0.789	-3.774	0.513	0.436	7.435	6.021
CaiT (XXS-24/16) (Touvron et al. 2021b)	93.28	0.662	-3.784	0.532	0.379	7.488	5.690
MoCo(v3) (ViT-B/16) (Chen, Xie, and He 2021)	93.12	0.130	-1.275	0.796	0.390	8.594	5.654
DINO (ViT-B/16) (Caron et al. 2021)	92.84	0.408	-3.610	0.475	0.510	8.454	5.675
Supervised pretrained (ViT-S/16) (Dosovitskiy et al. 2021)	92.70	0.886	-3.482	0.505	0.528	6.863	5.515
DeiT (ViT-T/16) (Touvron et al. 2021a)	90.12	0.797	-3.813	0.471	0.336	7.186	5.365
Supervised pretrained (ViT-T/16) (Dosovitskiy et al. 2021)	80.14	1.047	-3.211	0.438	0.303	4.988	3.814
DINO (ViT-S/16) (Caron et al. 2021)	76.54	0.295	-0.728	0.818	0.182	6.868	3.525
Pearson's ρ with Acc.		0.102	-0.623	-0.395	0.856	0.721	**0.974**
Kendall's τ_K with Acc.		-0.033	0.253	-0.055	0.626	0.516	**0.802**

Table A.7. Metrics evaluated on ImageNet-1k dataset using pre-trained ViT models. Because of the computational budget, we exclude the two largest models.

Algorithm	Acc. (%)	Alignment ↓	Uniformity ↓	Tolerance ↑	Linear CKA ↑	$\hat{I}_{\text{SimCLR}}(X;Y)$	$\hat{I}_{\text{class}}(X;Y)$
Supervised pretrained (ViT-B/16) (Dosovitskiy et al. 2021)	78.93	0.563	-3.889	0.432	0.365	9.199	7.208
DeiT (ViT-B/16) (Touvron et al. 2021a)	78.34	0.842	-3.834	0.482	0.234	8.679	8.009
PiT-S (Heo et al. 2021)	76.81	0.820	-3.833	0.472	0.198	8.513	7.543
CaiT (XXS-36/16) (Touvron et al. 2021b)	75.67	0.637	-3.840	0.550	0.228	8.373	6.795
DeiT (ViT-S/16) (Touvron et al. 2021a)	75.59	0.789	-3.852	0.498	0.209	8.278	7.280
MoCo(v3) (ViT-B/16) (Chen, Xie, and He 2021)	75.51	0.130	-1.297	0.792	0.268	9.524	6.658
CaiT (XXS-24/16) (Touvron et al. 2021b)	74.09	0.661	-3.864	0.516	0.205	8.315	6.547
DINO (ViT-B/16) (Caron et al. 2021)	73.28	0.411	-3.646	0.465	0.375	9.367	6.598
Supervised pretrained (ViT-S/16) (Dosovitskiy et al. 2021)	72.85	0.889	-3.506	0.494	0.428	7.572	6.233
DeiT (ViT-T/16) (Touvron et al. 2021a)	68.67	0.791	-3.872	0.462	0.197	7.874	5.883
Supervised pretrained (ViT-T/16) (Dosovitskiy et al. 2021)	53.01	1.044	-3.203	0.437	0.267	5.474	3.741
DINO (ViT-S/16) (Caron et al. 2021)	51.11	0.157	-0.702	0.881	0.193	7.426	3.316
Pearson's ρ with Acc.		0.077	-0.561	-0.392	0.203	0.783	**0.977**
Kendall's τ_K with Acc.		-0.030	-0.364	-0.061	0.152	0.576	**0.848**

Acknowledgement

First of all, I would like to thank my advisor, Wonjong Rhee, who supported me throughout my graduate school life. For seven years, you have taught me how to deeply understand issues (not only in academics but also in the real world) and explain my opinions in easy-to-understand terms. Thanks to your constant attention and advice, I was able to improve myself and complete my studies. Based on your teachings, I will continue to make an effort to become a better researcher.

Besides my doctoral advisor, I would like to thank my committee members: Professor Nojun Kwak, Professor Kyogu Lee, Professor Bongwon Suh, and Professor Daeyoung Choi. They provided me with insightful comments and valuable suggestions while writing the thesis.

Furthermore, I appreciate all the other members of Deep Representation Learning research group (Applied Data Science Laboratory): Jeongyun Han, Daeyoung Choi, Eunjung Lee, Hyunghun Cho, Moonjung Eo, Yoonah Lee, Jaekeol Choi, Won Shin, Sedong Kim, Seungwook Kim, Changho Shin, Seungeun Rho, Yongjae Lee, Jungwook Shin, Euna Jung, Duhun Hwang, Jaeill Kim, Donghun Lee, Jangwon Suh, Suhyeon Kang, Jimyeong Kim, Jungwon Park, Jinwoo Park, Jungmin Ko, Sungjun Lim, and Wonseok Lee. They provided valuable feedback to improve this work. It was a great honor to have time with such great colleagues.

Lastly, above all, I express my great love and respect for my family; Happy Hong, Rosa, Laurence, Stella, Heejae, puppy Kongkong, and grandmother Anna. Also, a special thanks to my friends who gave me a lot of support, Dohee, Moonjung, Eunjung, Nakyung, and Minjin. If there was no support of them, I could not complete my studies. You believed in me throughout this process, and I owe everything to you.