RényiCL: Contrastive Representation Learning with Skew Rényi Divergence

Kyungmin Lee **Jinwoo Shin**
Korea Advanced Institute of Science and Technology (KAIST)
{kyungmnlee, jinwoos}@kaist.ac.kr

Abstract

Contrastive representation learning seeks to acquire useful representations by estimating the shared information between multiple views of data. Here, the choice of data augmentation is sensitive to the quality of learned representations: as harder the data augmentations are applied, the views share more task-relevant information, but also task-irrelevant one that can hinder the generalization capability of representation. Motivated by this, we present a new robust contrastive learning scheme, coined RényiCL, which can effectively manage harder augmentations by utilizing Rényi divergence. Our method is built upon the variational lower bound of Rényi divergence, but a naïve usage of a variational method is impractical due to the large variance. To tackle this challenge, we propose a novel contrastive objective that conducts variational estimation of a skew Rényi divergence and provide a theoretical guarantee on how variational estimation of skew divergence leads to stable training. We show that Rényi contrastive learning objectives perform innate hard negative sampling and easy positive sampling simultaneously so that it can selectively learn useful features and ignore nuisance features. Through experiments on ImageNet, we show that Rényi contrastive learning with stronger augmentations outperforms other self-supervised methods without extra regularization or computational overhead. Moreover, we also validate our method on other domains such as graph and tabular, showing empirical gain over other contrastive methods. The implementation and pre-trained models are available at [1].

1 Introduction

Recently, many AI studies are enamored by the power of contrastive methods in learning useful representations without any human supervision, e.g., image [1, 2, 3, 4, 5, 6], text [7, 8], audio [9, 10] and multimodal video [11, 12, 13]. The key components of contrastive learning can be summarized into two-fold: data augmentation and contrastive objectives. The data augmentation generates different views of data where the views share information that is relevant to the downstream tasks. The contrastive objective then enforces the representation to capture the shared information between the views by bringing the views from the same data together and pushing the views from different data away in the representation space. Therefore, the choice of data augmentation is critical in contrastive learning, cf., see [3, 14, 15]. If the views share insufficient information, the representations cannot learn sufficient features for downstream tasks. On the other hand, if there is too much information between the views, the views might share nuisance features that hamper the generalization.

In this paper, we propose a new contrastive learning objective using Rényi divergence [16], coined *RényiCL*, which can effectively handle the case when the views share excessive information. The Rényi divergence is a generalization of KL divergence that is defined with an additional parameter

[1] https://github.com/kyungmnlee/RenyiCL

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

called an order. As the order becomes higher, the Rényi divergence penalizes more when two distributions mismatch. Therefore, since the goal of contrastive representation learning is to train an encoder to discriminate between positives and negatives, we hypothesize that maximizing the Rényi divergence between positives and negatives could lead to more discriminative representations when the data augmentations are aggressive. Furthermore, our theoretical analysis shows that RényiCL weighs importance on easy positives and hard negatives by using a higher order.

To implement RényiCL, one can consider variational lower bound of Rényi divergence [17]. However, we show that a variational lower bound of Rényi divergence does not suffice for contrastive learning due to their large variance. To tackle this challenge, we first made a key observation that the existing contrastive objectives are a variational form of a skew KL divergence, and show that variational estimation of skew divergence induces low variance property. Inspired by this, we consider a variational lower bound of a skew Rényi divergence and use it to implement RényiCL.

Through experiments under ImageNet [18], RényiCL (with harder data augmentations) achieves 76.2% linear evaluation accuracy with only 300 training epochs, outperforming other self-supervised methods without using large batch size or extra computational overhead. Moreover, we also show that RényiCL achieves better performance on transfer learning and few-shot classification, compared to baselines. Finally, we demonstrate that RényiCL provides empirical gains over original contrastive learning methods on various domains including images, graphs, and tabular datasets.

2 Related Works

Since Chen et al. [3] emphasized that the usage of data augmentation is crucial for contrastive learning, a series of works proposed various contrastive learning frameworks that showed great empirical success on visual representation learning [2, 3, 4, 5, 6, 19]. Inspired by the empirical success, recent works focused on optimal view generation for both positive and negatives. Tian et al. [14] proposed the InfoMin principle for view generation, which states that the views should share minimal information that is relevant to the downstream tasks. Another line of research focuses on hard negative samples [20, 21, 22] to enhance the performance of contrastive learning. Especially, Robinson et al. [20] proposed to use importance sampling to learn with hard negative samples.

Meanwhile, others are interested in searching for different contrastive objectives by modeling with various probabilistic divergences between positives and negatives. Oord et al. [23] proposed Contrastive Predictive Coding (CPC) (as known as InfoNCE) that is widely used in various contrastive learning scheme such as [2, 3, 4]. They theoretically prove that CPC objective is a variational lower bound to the mutual information, later [24] proposed Multi-Label CPC (MLCPC) which is a tighter lower bound to the mutual information. Hjelm et al. [25] proposed DeepInfoMax which used Jensen-Shannon divergence for variational estimation and maximization of mutual information. Zbontar et al. [26] proposed Barlow-Twins, which is shown to be a Hilbert-Schmidt Independence Criterion that approximates a Maximum Mean discrepancy measure between positives and negatives [27]. Ozair et al. [28] proposed Wasserstein Predictive Coding, which is both lower bound to the Wasserstein distance and mutual information. Similar to our approach, Tsai et al. [29] proposed relative predictive coding that uses skew χ^2-divergence for contrastive learning, and empirically and theoretically show that variational estimation of skew χ^2-divergence leads to stable contrastive representation learning. Lastly, our work is similar to [30], which proposed robust contrastive learning objective that uses symmetric binary classification loss for contrastive learning so that it can deal with noisy views.

3 Preliminaries

3.1 Rényi divergence

We first formally introduce Rényi divergence, which is a family of probability divergences including the popular Kullback-Leibler (KL) divergence as a special case. Let P, Q be two probability distributions such that P is absolutely continuous with respect to Q, denoted by $P \ll Q$ (i.e., the Radon-Nikodym derivative dP/dQ exists). Then, the *Rényi divergence* [16, 31] of order $\gamma \in (0,1) \cup (1,\infty)$ is defined by

$$R_\gamma(P \,\|\, Q) \coloneqq \frac{1}{\gamma(\gamma-1)} \log \mathbb{E}_P \left[\left(\frac{dP}{dQ} \right)^{\gamma-1} \right].$$

It is intertwined with various f-divergences, e.g., Rényi divergence with $\gamma \to 1$ and $\gamma = 2$ becomes KL divergence $D_{\mathtt{KL}}(P \,\|\, Q)$ and a monotonic transformation of χ^2 divergence, respectively. Remark that the Rényi divergence with higher order γ, penalizes more when the probability mass of P does not overlap with Q [32]. Rényi divergence has been studied in various machine learning tasks such as variational inference [33, 34] and training [35, 36] or evaluation of generative models [37].

3.2 Variational estimation of divergences and mutual information

We now review the variational estimators of probability divergences associated with neural networks. Donsker-Varadhan (DV) [38] objective is a variational representation of KL divergence defined as follows:

$$D_{\mathtt{KL}}(P \,\|\, Q) = \sup_{f \in \mathcal{F}} \mathcal{I}_{\mathtt{DV}}(f), \quad \text{for} \quad \mathcal{I}_{\mathtt{DV}}(f) \coloneqq \mathbb{E}_P[f] - \log \mathbb{E}_Q[e^f], \tag{1}$$

where $\mathcal{F}$ is a set of bounded measurable functions on the support of P and Q. Remark that the optimum $f^* \in \mathcal{F}$ of (1) satisfies $f^* \propto \log \frac{dP}{dQ}$.

One often considers variational estimator of *mutual information* where it is defined by the KL divergence between the joint distribution and the product of marginal distributions. Formally, given two random variables X, Y, let P_{XY} be the joint distribution of $X \times Y$, and $P_X P_Y$ be the product of marginal distributions over $X \times Y$. Then, we call the pair of samples $(x, y) \sim P_{XY}$ as a 'positive pair', and $(x, y) \sim P_X P_Y$ as a 'negative pair'. Then, the mutual information is defined by

$$\mathcal{I}(X; Y) \coloneqq D_{\mathtt{KL}}(P_{XY} \,\|\, P_X P_Y).$$

Since the mutual information is intractable to compute unless the densities of X and Y are explicitly known, many works resort to optimize its variational lower bounds [39] associated with neural networks [40]. Belghazi et al. [40] used the DV objective (1) to compute mutual information, i.e.,

$$\mathcal{I}_{\mathtt{DV}}(f) = \mathbb{E}_{P_{XY}}[f(x, y)] - \log \mathbb{E}_{P_X P_Y}[e^{f(x,y)}].$$

However, Song et al. [41] showed that the above objective might have large variance unless one uses large number of samples, and Tsai et al. [29] also empirically evidenced that contrastive learning with the DV objective suffers from training instability.

To address the issue, the *contrastive predictive coding* (CPC) objective (also as known as InfoNCE) [23] is a popular choice for various practices including contrastive representation learning [2, 4, 3]. Given B batch of samples $\{x_i\}_{i=1}^B$ from X, assume we have a single positive y_i^+ and K negatives $\{y_{ij}^-\}_{j=1}^K$ for each x_i. Then, the CPC objective is defined as follows:

$$\mathcal{I}_{\mathtt{CPC}}(f) \coloneqq \mathbb{E}\left[\frac{1}{B}\sum_{i=1}^{B} \log \frac{(K+1) \cdot e^{f(x_i, y_i^+)}}{e^{f(x_i, y_i^+)} + \sum_{j=1}^K e^{f(x_i, y_{ij}^-)}}\right],$$

where it is known that $\mathcal{I}_{\mathtt{CPC}}(f) \leq \mathcal{I}(X; Y)$ for any bounded measurable function f [23]. However, as $\mathcal{I}_{\mathtt{CPC}}(f) \leq \log(K+1)$ for any f, the CPC objective becomes a high-bias estimator if the true value $\mathcal{I}(X; Y)$ is larger than $\log(K+1)$. To address this, Poole et al. [39] proposed α-CPC which controls the bias by inserting $\alpha \in [0, 1]$ into CPC as follows:

$$\mathcal{I}_{\mathtt{CPC}}^{(\alpha)}(f) \coloneqq \mathbb{E}\left[\frac{1}{B}\sum_{i=1}^{B} \log \frac{e^{f(x_i, y_i^+)}}{\alpha e^{f(x_i, y_i^+)} + \frac{1-\alpha}{K}\sum_{j=1}^K e^{f(x_i, y_{ij}^-)}}\right]. \tag{2}$$

The α-CPC objective also admits a variational lower bound to the mutual information $\mathcal{I}(X; Y)$ for any $\alpha \in [0, 1]$ (recovers the original CPC when $\alpha = \frac{1}{K+1}$), and it achieves smaller bias when using smaller value of α. Furthermore, Song et al. [24] proposed α-*Multi Label CPC* (MLCPC) which provides even a tighter lower bound to the mutual information defined as follows:

$$\mathcal{I}_{\mathtt{MLCPC}}^{(\alpha)}(f) = \mathbb{E}\left[\frac{1}{B}\sum_{i=1}^{B} \log \frac{e^{f(x_i, y_i^+)}}{\frac{\alpha}{B}\sum_{i=1}^B e^{f(x_i, y_i^+)} + \frac{1-\alpha}{BK}\sum_{i=1}^B\sum_{j=1}^K e^{f(x_i, y_{ij}^-)}}\right]. \tag{3}$$

4 Variational Estimation of Skew Rényi Divergence

Similar to variational estimators of $D_{\mathrm{KL}}(P_{XY} \| P_X P_Y)$ in the previous section, we consider a variational estimator of Rényi divergence $R_\gamma(P_{XY} \| P_X P_Y)$, in particular, considering its application to contrastive representation learning. To that end, we first introduce the following known lemma which states variational representation of Rényi divergence similar to the DV objective in (1).

Lemma 4.1 ([17]). *For distributions P, Q such that $P \ll Q$, let $\mathcal{F}$ be a set of bounded measurable functions. Then Rényi divergence of order $\gamma \in (0,1) \cup (1,\infty)$ admits following variational form:*

$$R_\gamma(P \,\|\, Q) = \sup_{f \in \mathcal{F}} \mathcal{I}^{(\gamma)}_{\textit{Rényi}}(f) \quad \textit{for} \quad \mathcal{I}^{(\gamma)}_{\textit{Rényi}}(f) \coloneqq \frac{1}{\gamma - 1} \log \mathbb{E}_P[e^{(\gamma-1)f}] - \frac{1}{\gamma} \log \mathbb{E}_Q[e^{\gamma f}], \tag{4}$$

where the optimum $f^* \in \mathcal{F}$ satisfies $f^* \propto \log \frac{dP}{dQ}$.

By continuity, $\mathcal{I}^{(\gamma)}_{\text{Rényi}}(f)$ in (4) equals to DV objective (1) when $\gamma \to 1$.

4.1 Challenges in variational Rényi divergence estimation

To perform contrastive representation learning, one can use the variational estimator (4) for $R_\gamma(P_{XY} \| P_X P_Y)$, similarly as did for KL divergence. However, the contrastive learning with (4) is impractical due to the large variance. In Appendix C.2, we show that even for a simple synthetic Gaussian dataset, (4) is not able to estimate Rényi divergence between two correlated Gaussian distributions if the correlation coefficient is large. The large-variance issue of (4) resembles that of the variational mutual information estimator well observed in the literature [29, 39, 41]. In the following theorem, we provide an analogous result for the Rényi variational objective (4).

Theorem 4.1. *Assume $P \ll Q$, and $\mathit{Var}_Q[dP/dQ] < \infty$. Let P_m and Q_n be the empirical distributions of m i.i.d samples from P and n i.i.d samples from Q, respectively. Define*

$$\hat{\mathcal{I}}^{(\gamma)}_{\textit{Rényi}}(f) \coloneqq \frac{1}{\gamma - 1} \log \mathbb{E}_{P_m}[e^{(\gamma-1)f}] - \frac{1}{\gamma} \log \mathbb{E}_{Q_n}[e^{\gamma f}],$$

and assume we have $f^* \propto \log(dP/dQ)$. Then $\forall \gamma > 1, \forall m \in \mathbb{N}$, we have

$$\lim_{n \to \infty} n \cdot \mathit{Var}_{P,Q}[\hat{\mathcal{I}}^{(\gamma)}_{\textit{Rényi}}(f^*)] \geq \frac{e^{\gamma^2 D_{\mathrm{KL}}(P \,\|\, Q)} - \gamma^2}{e^{2\gamma(\gamma-1) R_\gamma(P \,\|\, Q)}}.$$

The proof of Theorem 4.1 is in Appendix A.1. Theorem 4.1 implies that even though one achieves the optimal function for variational estimation, the variance of (4) could explode exponentially with respect to the ground-truth mutual information. This result is coherent with [42], which identified the problems in variational estimation of mutual information.

On the other hand, we recall that CPC and MLCPC are empirically shown to be low variance estimators [39, 24]. Then, the natural question arises: what makes CPC and MLCPC fundamentally different from the DV objective? In the following section, we answer to this question by showing that the CPC and MLCPC objectives are variational lower bounds of a skew KL divergence, and provide a theoretical evidence that variational estimators of skew divergence can have low variance. This insight will be used later for designing a low variance estimator for the desired Rényi divergence.

4.2 Low variance of variational skew divergence estimators

We first define α-skew KL divergence. For distributions P, Q with $P \ll Q$ and for any $\alpha \in [0,1]$, the α-skew KL divergence between P and Q is defined by the KL divergence between P and the mixture $\alpha P + (1-\alpha)Q$:

$$D^{(\alpha)}_{\mathrm{KL}}(P \,\|\, Q) \coloneqq D_{\mathrm{KL}}(P \,\|\, \alpha P + (1-\alpha)Q).$$

One can see that the DV objective (1) for α-skew KL divergence can be written as following:

$$D^{(\alpha)}_{\mathrm{KL}}(P \,\|\, Q) = \sup_{f \in \mathcal{F}} \mathbb{E}_P[f] - \log\left(\alpha \mathbb{E}_P[e^f] + (1-\alpha)\mathbb{E}_Q[e^f]\right). \tag{5}$$

We claim that (5) is equivalent to α-CPC and α-MLCPC. Remark that the generalized form of α-CPC objective (2) can be written as follows:

$$\mathcal{I}_{\texttt{CPC}}^{(\alpha)}(f) = \mathbb{E}_{P_{XY}}[f(x,y)] - \mathbb{E}_{P_X}\left[\log\left(\alpha\mathbb{E}_{P_{Y|X}}[e^{f(x,y)}] + (1-\alpha)\mathbb{E}_{P_Y}[e^{f(x,y)}]\right)\right]. \tag{6}$$

Then, one can observe that (6) is equivalent to the expectation of the variational DV objective (5) of α-skew KL divergence between $P_{Y|X}$ and P_Y over P_X, which shows that α-CPC is a DV objective of α-skew KL divergence between P_{XY} and $P_X P_Y$:

$$\sup_{f\in\mathcal{F}} \mathcal{I}_{\texttt{CPC}}^{(\alpha)}(f) = \mathbb{E}_{P_X}[D_{\texttt{KL}}^{(\alpha)}(P_{Y|X} \,\|\, P_Y)] = D_{\texttt{KL}}^{(\alpha)}(P_{X,Y} \,\|\, P_X P_Y).$$

Similarly, the generalized form of α-MLCPC objective (3) is given by

$$\mathcal{I}_{\texttt{MLCPC}}^{(\alpha)}(f) = \mathbb{E}_{P_{XY}}[f(x,y)] - \log\left(\alpha\mathbb{E}_{P_{XY}}[e^{f(x,y)}] + (1-\alpha)\mathbb{E}_{P_X P_Y}[e^{f(x,y)}]\right),$$

which also becomes the DV objective for α-skew KL divergence $D_{\texttt{KL}}^{(\alpha)}(P_{XY}\|P_X P_Y)$.

Further, we formally show that the variational estimation of empirical skew divergence has low variance, explaining why CPC and MLCPC objectives become low-variance estimators of the mutual information. The following theorem shows that the variance of the variational estimator can be adjusted with the right choice of α.

Theorem 4.2. *Assume $P \ll Q$ and $Var_Q[dP/dQ] < \infty$. Let P_m and Q_n be empirical distributions of m i.i.d samples from P and n i.i.d samples from Q. Then define*

$$\hat{\mathcal{I}}_{\texttt{KL}}^{(\alpha)}(f) = \mathbb{E}_{P_m}[f] - \log\left(\alpha\mathbb{E}_{P_m}[e^f] + (1-\alpha)\mathbb{E}_{Q_n}[e^f]\right),$$

and assume that there is $\hat{f} \in \mathcal{F}$ that $|\mathcal{I}_{\texttt{KL}}^{(\alpha)}(\hat{f}) - D_{\texttt{KL}}^{(\alpha)}(P_m \,\|\, Q_n)| < \varepsilon_f$ for some $\varepsilon_f > 0$. Then for $\forall\alpha < 1/8$, the variance of estimator satisfies

$$Var_{P,Q}[\hat{\mathcal{I}}_{\texttt{KL}}^{(\alpha)}(\hat{f})] \le c_1\varepsilon_f + \frac{c_2(\alpha)}{\min\{n,m\}} + \frac{c_3\log^2(\alpha m)}{m} + \frac{c_4\log^2(c_5 n)}{\alpha^2 n},$$

for some constants $c_1, c_3, c_4, c_5 > 0$ that are independent of n, m, α, and $D_{\texttt{KL}}(P\|Q)$, and $c_2(\alpha)$ satisfies $c_2(\alpha) = \min\{\frac{1}{\alpha}, \frac{\chi^2(P\|Q)}{1-\alpha}\}$, where $\chi^2(P\|Q) = \mathbb{E}_Q[(dP/dQ)^2]$.

The proof is in Appendix A.2. Since we have $\chi^2(P\|Q) \ge e^{D_{\texttt{KL}}(P\|Q)} - 1$, if α is too small, the bound in Theorem 4.2 is loose as in Theorem 4.1. Therefore, Theorem 4.2 implies that one should use sufficiently large α to achieve low variance. Theorem. 4.2 demonstrates that if one can find a sufficiently close critic for empirical skew KL divergence, the variance of the estimator is asymptotically bounded unless we choose large enough $\alpha = \alpha_n = \omega(n^{-1/2})$.

4.3 Variational estimation of skew Rényi divergence

Inspired by the theoretical guarantee that skew divergence can achieve bounded variance, we now present a variational estimator of skew Rényi divergence. Remark that the analogous version of Theorem 4.2 can be achieved for Rényi divergence (see Appendix A.3). For any $\alpha \in [0,1]$ and $\gamma \in (0,1)\cup(1,\infty)$, the α-skew Rényi divergence of order γ is defined as:

$$R_\gamma^{(\alpha)}(P \,\|\, Q) \coloneqq R_\gamma(P \,\|\, \alpha P + (1-\alpha)Q).$$

Then, we define (α,γ)-*Rényi Multi-Label CPC* (RMLCPC) objective by using the variational lower bound (4) for α-skew Rényi divergence between P_{XY} and $P_X P_Y$:

$$\mathcal{I}_{\texttt{RMLCPC}}^{(\alpha,\gamma)}(f) \coloneqq \frac{1}{\gamma-1}\log\mathbb{E}_{P_{XY}}[e^{(\gamma-1)f(x,y)}] - \frac{1}{\gamma}\log(\alpha\mathbb{E}_{P_{XY}}[e^{\gamma f(x,y)}] + (1-\alpha)\mathbb{E}_{P_X P_Y}[e^{\gamma f(x,y)}]),$$

which satisfies $\sup_f \mathcal{I}_{\texttt{RMLCPC}}^{(\alpha,\gamma)}(f) = R_\gamma^{(\alpha)}(P_{XY} \,\|\, P_X P_Y)$ for any $\alpha \in [0,1]$ and $\gamma \in (0,1)\cup(1,\infty)$.

Note that the RMLCPC objective can be used for contrastive learning with multiple positive views. For example, when using multi-crops data augmentation [43], instead of computing contrastive objective for each crop, we gather all positive pairs and negative pairs and compute only once for the final loss. The PyTorch style pseudo-code for the RMLCPC objective is in Algorithm 1.

5 Rényi Contrastive Representation Learning

In this section, we present *Rényi contrastive learning* (RényiCL) where we use the RMLCPC objective for contrastive representation learning. As we discussed earlier, the Rényi divergence penalizes more when two distributions differ. Thus, when using harder data augmentations in contrastive learning, one can expect that RényiCL can learn more discriminative representation.

5.1 Rényi contrastive representation learning

We begin with backgrounds for contrastive representation learning. Given a dataset $\mathcal{X}$, the goal of representation learning is to train an encoder $g : \mathcal{X} \to \mathbb{R}^d$ that is a useful feature of $\mathcal{X}$. Especially, contrastive representation learning enforces g to discriminate between positive pairs and negative pairs [2], where positive pairs are generated by applying data augmentation on the same data and negative pairs are augmented data from different source data. Formally, let V, V' be random variables for augmented views from dataset $\mathcal{X}$. Then denote $P_{VV'}$ be the joint distribution of a positive pair and $P_V P_{V'}$ be the distribution of negative pairs. For the sake of brevity, we denote $(z, z^+) \sim P_{VV'}$ be a positive pair, and $(z, z^-) \sim P_V P_{V'}$ be a negative pair. Also, let $z = g(v), z' = g(v')$ be features of the encoder and $Z = g(V), Z' = g(V')$ be corresponding random variables of feature distributions. Then we define $(z, z^+) \sim P_{ZZ'}$ be features of positive views, and $(z, z^-) \sim P_Z P_{Z'}$ be features of negative views.

The InfoMax principle [25] for representation learning aims to find g that preserves the maximal mutual information between V and V' by following:

$$\sup_{g:\mathcal{X}\to\mathbb{R}^d} D_{\mathtt{KL}}(P_{ZZ'} \,\|\, P_Z P_{Z'}) = \sup_{g:\mathcal{X}\to\mathbb{R}^d} \mathcal{I}\left(Z; Z'\right) \leq \mathcal{I}(V; V'),$$

i.e., finds a neural encoder g which discriminates between positive and negative pairs as much as possible. Here, α-CPC or α-MLCPC are used for plug-in variational estimators of mutual information, for example contrastive learning with α-MLCPC satisfies following:

$$\sup_{g:\mathcal{X}\to\mathbb{R}^d} \sup_{f\in\mathcal{F}} \mathcal{I}^{(\alpha)}_{\mathtt{MLCPC}}(f, g) = \sup_{g:\mathcal{X}\to\mathbb{R}^d} D^{(\alpha)}_{\mathtt{KL}}(P_{ZZ'} \| P_Z P_{Z'}),$$

since $\mathcal{I}^{(\alpha)}_{\mathtt{MLCPC}}(f, g)$ is a variational estimator of $D^{(\alpha)}_{\mathtt{KL}}(P_{ZZ'} \| P_Z P_{Z'})$.

Now we present Rényi contrastive representation learning, which considers the following optimization problem by using skew Rényi divergence:

$$\sup_{g:\mathcal{X}\to\mathbb{R}^d} \sup_{f\in\mathcal{F}} \mathcal{I}^{(\alpha,\gamma)}_{\mathtt{RMLCPC}}(f, g) = \sup_{g:\mathcal{X}\to\mathbb{R}^d} R^{(\alpha)}_{\gamma}(P_{ZZ'} \| P_Z P_{Z'}),$$

since $\mathcal{I}^{(\alpha,\gamma)}_{\mathtt{RMLCPC}}(f, g)$ is a variational estimator of $R^{(\alpha)}_{\gamma}(P_{ZZ'} \| P_Z P_{Z'})$.

5.2 Gradient analysis for RényiCL

In this section, we provide an analysis on the effect of γ in RényiCL based on the gradient of contrastive objectives. For simplicity, we consider the case when $\alpha = 0$, and we provide general analysis for $\alpha > 0$ in Appendix B.1. Let f_θ be a neural network parameterized by θ. By using the reparametrization trick, the gradient of $\mathcal{I}_{\mathtt{MLCPC}}(f_\theta)$ becomes

$$\begin{aligned}\nabla_\theta \mathcal{I}_{\mathtt{MLCPC}}(f_\theta) &= \mathbb{E}_{z,z^+}[\nabla_\theta f_\theta(z, z^+)] - \frac{\mathbb{E}_{z,z^-}[e^{f_\theta(z,z^-)} \nabla_\theta f_\theta(z, z^-)]}{\mathbb{E}_{z,z^-}[e^{f_\theta(z,z^-)}]} \\ &= \mathbb{E}_{z,z^+}[\nabla_\theta f_\theta(z, z^+)] - \mathbb{E}_{\mathtt{sg}(q_\theta(z,z^-))}[\nabla_\theta f_\theta(z, z^-)],\end{aligned}$$

where $q_\theta(z, z^-) \propto e^{f_\theta(z,z^-)}$ is a self-normalized importance weights [44], and $\mathtt{sg}$ is a stop-gradient operator. Thus, the MLCPC objective is equivalent to following in terms of gradient [45]:

$$\mathcal{I}_{\mathtt{MLCPC}}(f_\theta) = \mathbb{E}_{z,z^+}[f_\theta(z, z^+)] - \mathbb{E}_{\mathtt{sg}(q_\theta(z,z^-))}[f_\theta(z, z^-)]. \tag{7}$$

This shows that the original contrastive objective performs innate hard negative sampling with importance weight q_θ [20], as the gradient of negative pairs becomes larger as the value of f_θ

Table 1: Linear evaluation on the ImageNet validation set. We report pre-training epochs and Top-1 classification accuracies (%).

Method	Epochs	Top-1
SimCLR [3]	800	70.4
Barlow Twins [26]	800	73.2
BYOL [47]	800	74.3
MoCo v3 [6]	800	74.6
SwAV [43]	800	75.3
DINO [43]	800	75.3
NNCLR [19]	1000	75.6
C-BYOL [48]	1000	75.6
RényiCL	**200**	**75.3**
RényiCL	**300**	**76.2**

Table 2: Semi-supervised learning results on ImageNet. We report Top-1 and Top-5 classification accuracies (%) by fine-tuning a pre-trained ResNet-50 with 1% and 10% ImageNet datasets.

	1% ImageNet		10% ImageNet	
Method	Top-1	Top-5	Top-1	Top-5
Supervised [3]	25.4	48.4	56.4	80.4
SimCLR [3]	48.3	75.5	65.6	87.8
BYOL [47]	53.2	78.4	68.8	89.0
SwAV [43]	53.9	78.5	70.2	89.9
Barlow Twins [26]	55.0	79.2	69.7	89.3
NNCLR [19]	56.4	80.7	69.8	89.3
C-BYOL [48]	**60.6**	**83.4**	70.5	90.0
RényiCL	56.4	80.6	**71.2**	**90.3**

becomes larger. On the other hand, for RMLCPC objective, by letting $\alpha = 0$, and taking gradient with respect to θ, we have

$$\begin{aligned}\nabla_\theta \mathcal{I}^{(\gamma)}_{\texttt{RMLCPC}}(f_\theta) &= \frac{\mathbb{E}_{z,z^+}[e^{(\gamma-1)f_\theta(z,z^+)}\nabla_\theta f_\theta(z,z^+)]}{\mathbb{E}_{z,z^+}[e^{(\gamma-1)f_\theta(z,z^+)}]} - \frac{\mathbb{E}_{z,z^-}[e^{\gamma f_\theta(z,z^-)}\nabla_\theta f_\theta(z,z^-)]}{\mathbb{E}_{z,z^-}[e^{\gamma f_\theta(z,z^-)}]} \\ &= \mathbb{E}_{\texttt{sg}(q_\theta(z,z^+;\gamma-1))}[\nabla_\theta f_\theta(z,z^+)] - \mathbb{E}_{\texttt{sg}(q_\theta(z,z^-;\gamma))}[\nabla_\theta f_\theta(z,z^-)],\end{aligned}$$

where $\texttt{sg}(q_\theta(z,z^+;\gamma-1)) \propto e^{(\gamma-1)f_\theta(z,z^+)}$ and $\texttt{sg}(q_\theta(z,z^-;\gamma)) \propto e^{\gamma f_\theta(z,z^-)}$ are self-normalizing importance weights for each positive and negative term. Hence, the RMLCPC objective is equivalent to the following in terms of gradient:

$$\mathcal{I}^{(\gamma)}_{\texttt{RMLCPC}}(f_\theta) = \mathbb{E}_{\texttt{sg}(q_\theta(z,z^+;\gamma-1))}[f_\theta(z,z^+)] - \mathbb{E}_{\texttt{sg}(q_\theta(z,z^-;\gamma))}[f_\theta(z,z^-)]. \tag{8}$$

Remark that (7) and (8) have common form that f_θ is maximized for positive pairs and minimized for negative pairs, which is similar to contrastive divergence [46]. On the other hand, (7) and (8) have two differences:

- **Hard negative sampling**: the gradient weighs more on harder negatives, i.e., (z, z^-) with high value of $f_\theta(z, z^-)$, as γ increases [20].
- **Easy positive sampling**: the gradient weighs more on easier positives, i.e., (z, z^+) with high value of $f_\theta(z, z^+)$ as $\gamma \in (1, \infty)$ increases.

Therefore, when there are positive views that share task-irrelevant information, the RMLCPC objective resists updating, and it regularizes the model to ignore task-irrelevant information. The hard negative sampling with importance weight was proposed in [20], but our analysis shows that CPC and MLCPC conduct intrinsic hard negative sampling. Also, the Rényi contrastive learning can control the level of hard negative sampling by choosing the appropriate γ. Lastly, remark that our analysis is given for $\alpha = 0$, but in practice we use nonzero value of α. In Appendix B.1, we show that it requires sufficiently small values of α to have the effect of easy positive sampling for harder data augmentations.

6 Experiments

6.1 RényiCL for visual representation learning on ImageNet

Setup. For ImageNet [18] experiments, we use ResNet-50 [49] for encoder g. We use MLP projection head with momentum encoder [4], which is updated by EMA, and we use predictor, following the practice of [47, 6]. Then the critic f is implemented by the cosine similarity between the output of momentum encoder and base encoder with predictor, divided by temperature $\tau = 0.5$. We maximize RMLCPC objective with $\gamma = 2.0$.

Data augmentation. Given the popular data augmentation baseline [47, 3], we further use RandAugment [50] and multi-crop augmentation [43] to implement harder data augmentation. When using

Table 3: Transfer learning performance on object classification datasets. For Aircraft and Flowers, we report mean per-class accuracy (%), and for VOC2007, we report 11-point mAP. Otherwise, we report Top-1 classification accuracies (%).

Method	CIFAR10	CIFAR100	Food101	Flowers	Cars	Aircraft	DTD	SUN397	VOC2007
Supervised [3]	93.6	78.3	72.3	94.7	66.7	61.0	74.9	61.9	82.8
SimCLR [3]	90.6	71.6	68.4	91.2	50.3	50.3	74.5	58.8	81.4
BYOL [47]	91.3	78.4	75.3	96.1	67.8	60.6	75.5	62.2	82.5
NNCLR [19]	93.7	**79.0**	76.7	95.1	67.1	**64.1**	75.5	62.5	83.0
RényiCL	**94.4**	**79.0**	**78.0**	**96.5**	**71.5**	61.8	**77.3**	**66.1**	**88.2**

Table 4: Comparison on contrastive self-supervised methods with harder augmentations. We compare Top-1 linear evaluation accuracy (%) on ImageNet, Pets, Caltech101 datasets and few-shot classification accuracy (%) over 2000 episodes on FC100, CUB200, and Plant disease datasets. † denotes trained without multi-crops. Here, we consider RényiCL without multi-crops as other baselines do, i.e., for fair comparison.

		Linear evaluation			5-way 1-shot			5-way 5-shot		
Method	Epochs	ImageNet	Pets	Caltech101	FC100	CUB200	Plant	FC100	CUB200	Plant
InfoMin [14]	800	**73.0**	86.37	88.48	31.80	53.81	66.11	45.09	72.20	84.12
CLSA [59]	800	72.2	85.18	91.21	34.46	50.83	67.39	49.16	67.93	86.15
RényiCL†	200	72.6	**88.43**	**94.04**	**36.31**	**59.73**	**80.68**	**53.39**	**82.12**	**94.29**

multi-crop, we gather all positives and negatives, then directly compute the RMLCPC objective, which differs from the original method in [43] (see 1 in Appendix B.1).

Linear evaluation. We follow the linear evaluation protocol, where we report the Top-1 ImageNet validation accuracy (%) of a linear classifier trained on the top of frozen features. Table 1 compares the performance of different self-supervised methods using linear evaluation protocol. First, RényiCL outperforms other self-supervised methods with a large margin by only training for 300 epochs. Meanwhile, RényiCL enjoys better computational efficiency in that it does not require a large batch size [3, 47, 43, 6] or extra memory queue [19].

Semi-supervised learning. We evaluate the usefulness of the learned feature in a semi-supervised setting with 1% and 10% subsets of ImageNet dataset [3, 43]. The results are in Table 2. We observe that RényiCL achieves the best performance on fine-tuning with 10% subset, and runner-up on fine-tuning with 1% subset, showing the generalization capacity in low-shot learning scenarios.

Transfer learning. We evaluate the learned representation through the performance of transfer learning on fine-grained object classification datasets by using linear evaluation. We use CIFAR10/100 [51], Food101 [52], Flowers [53], Cars [54], Aircraft [55], DTD [56], SUN397 [57], and VOC2007 [58]. In Table 3, we compare it with other self-supervised methods. Note that RényiCL achieves the best performance in 8 out of 9 datasets, especially showing superior performance on SUN397 (3.6%) and VOC2007 (5.2%) datasets.

Comparison among self-supervised methods using hard augmentations. To demonstrate the effectiveness of RényiCL on learning with harder augmentations, we compare with InfoMin [14] and CLSA [59] where they also use harder augmentations for contrastive representation learning. Table 4 demonstrates the performance of representations learned with different self-supervised methods by linear evaulation on ImageNet [18], Pets [60], Caltech101 [61], and few-shot classification on FC100 [62], Caltech-UCSD Birds (CUB200) [50], and Plant Disease [63] datasets. While RényiCL achieves slightly lower performance than InfoMin on ImageNet validation accuracy, it outperforms InfoMin and CLSA on other transfer learning tasks with a large margin, which indicates the superiority of RényiCL on learning better representations under the hard data augmentations.

6.2 RényiCL for representation learning on various domains

Setup. We perform ablation studies of RényiCL on various domains such as images, tabular, and graph datasets. For images, we consider CIFAR-10/100 [51] and ImageNet-100 [64], which is a 100-

Table 5: Ablation on contrastive objectives for unsupervised representation learning on image (ImageNet-100, CIFAR-100, CIFAR-10) and tabular (CovType, Higgs-100K) datasets. We report Top-1 accuracy (%) with linear evaluation. Average over 5 runs.

	ImageNet-100		CIFAR-100		CIFAR-10		CovType		Higgs-100K	
Method	Base	Hard	Base	Hard	Base	Hard	Base	Hard	Base	Hard
CPC	79.7	81.1(+1.4)	65.4	67.1(+1.7)	91.7	91.9(+0.2)	71.6	74.3(+2.7)	64.7	71.3(+6.6)
MLCPC	79.5	81.2(+1.7)	65.6	66.6(+1.0)	91.9	92.1(+0.2)	71.7	74.1(+2.4)	64.9	71.5(+6.6)
RMLCPC	78.9	**81.6(+2.7)**	64.5	**68.5(+4.0)**	90.7	**92.5(+1.8)**	72.1	**74.9(+2.8)**	64.5	**72.4(+7.9)**

Table 6: Comparison of different contrastive learning methods on graph TUDataset. Average over 5 runs.

Method	NCI1	PROTEINS	DD	MUTAG	COLLAB	RDT-B	RDT-M5K	IMDB-B
InfoGraph [72]	76.20	74.44	72.85	89.01	70.65	82.50	53.46	**73.03**
GraphCL [15]	77.87	74.39	78.62	86.80	71.36	89.53	55.99	71.14
JOAO [73]	78.07	74.55	77.32	87.35	69.50	85.29	55.74	70.21
JOAOv2 [73]	78.36	74.07	77.40	87.67	69.33	86.42	56.03	70.83
RényiCL	**78.60**	**75.11**	**78.98**	**90.22**	**71.88**	**90.92**	**56.18**	72.38

class subset of ImageNet [18]. For the base data augmentation, we use a popular benchmark from [3], and for the harder augmentation, we further apply RandAugment [50] and RandomErasing [65]. For ImageNet-100, we use the default settings of [6] with ResNet-50 backbone with only change in learning objective. For CIFAR-10 and CIFAR-100, we follow the settings in [3] with ResNet-18 backbone adjusted for the CIFAR dataset. We use same α for all CPC, MLCPC, RMLCPC, and use $\gamma = 2.0$ for RMLCPC.

For tabular experiments, we use Forest Cover Type (CovType) and Higgs Boson (Higgs) [66] dataset from UCI repository [67]. Due to its massive size, we consider a subset of 100K for the Higgs experiments. Since the tabular dataset has limited domain knowledge, we do not use any data augmentation for the baseline and when using data augmentation, we use random masking noise [68] for the Higgs dataset and random feature corruption [69] for the CovType dataset. We follow the settings of MoCo v3 [6] except that a 5-layer MLP is used for the backbone. We use $\gamma = 1.1$ for CovType and $\gamma = 1.2$ for Higgs. For evaluation, we use linear evaluation for CovType and we compute the linear regression layer by pseudo-inverse between the feature matrix and label matrix.

Lastly, we examine RényiCL on graph TUDataset [70], which contains numerous graph machine learning benchmarks of different domains such as bioinformatics, molecules, and social network. We follow the experimental setup of [15]: we use graph isomorphism network [71] backbone and use node dropout, edge perturbation, attribute masking, and subgraph sampling for data augmentations. For the graph experiments, we do not change the strength of data augmentation nor add new data augmentation. The detailed hyper-parameters for training and evaluation for the whole dataset can be found in Appendix B.3.

Results. Table 5 compares the Top-1 linear evaluation accuracy (%) of representations with different contrastive learning objectives. We observe that RMLCPC obtains the best performance when using harder data augmentations on all images and tabular datasets. Also, RMLCPC has the largest gain on using harder augmentation, showing that the effectiveness of RényiCL is amplified when we use stronger augmentation. Table 6 compare the Top-1 linear evaluation accuracy (%) of graph RényiCL with other graph contrastive learning methods. Even without harder data augmentation, RényiCL achieves the best performance in 7 out of 8 benchmarks. Especially, RényiCL outperforms JOAO [73], where the views are learned to boost the performance of graph contrastive learning.

6.3 Mutual information estimation and view selection

Remark that CPC and MLCPC are intrinsically high-bias mutual information estimator as α-skew KL divergence is strictly smaller than original KL divergence for $0 < \alpha < 1$, i.e.,

$$D_{\mathtt{KL}}^{(\alpha)}(P\|Q) \leq (1-\alpha)D_{\mathtt{KL}}(P\|Q) < D_{\mathtt{KL}}(P\|Q),$$

from the convexity of KL divergence. Similarly, RMLCPC is also a strictly-biased estimator of Rényi divergence. However, one can recover the original mutual information by using the optimal

condition of function f from all CPC, MLCPC, and RMLCPC objectives [29]. Here, we describe the sketch and present detailed method in Appendix C.1. From the optimality condition of (1) and (4), the optimal critic f^* of α-CPC, α-MLCPC and (α, γ)-RMLCPC of any γ satisfies $f^* \propto \log \frac{dP_{XY}}{\alpha dP_{XY} + (1-\alpha) dP_X dP_Y}$, then we use Monte-Carlo method [44] to approximate log-normalization constant, and use it to approximate the log-density ratio $\hat{r} = \log \frac{dP_{XY}}{dP_X dP_Y}$. Then we estimate the mutual information by $\hat{\mathcal{I}}(X;Y) = \frac{1}{B} \sum_{i=1}^{B} \log \hat{r}(x_i, y_i)$, where $(x_i, y_i) \sim P_{XY}$ are positive pairs.

Mutual information estimation of synthetic Gaussian. Following [39, 24], we conduct experiments on estimating the mutual information of multivariate Gaussian distributions. In Appendix C.2, we show that while α-CPC, α-MLCPC, and (α, γ)-RMLCPC objectives are high biased objectives, one can achieve low bias, low variance estimator by using the approximation method above.

7 Conclusion and Discussion

We propose Rényi Contrastive Learning (RényiCL) which utilizes Rényi divergence to deal with stronger data augmentations. Since the variational lower bound of Rényi divergence is insufficient for contrastive learning due to large variance, we introduce a variational lower bound of skew Rényi divergence, namely Rényi-MLCPC. Indeed, we show that CPC and MLCPC are variational forms of skew KL divergence, and provide theoretical analysis on how they achieve low variance. Through experiments, we validate the effectiveness of RényiCL by using harder data augmentations.

Limitations and future works. While RényiCL is beneficial when using harder data augmentations, we did not identified the optimal data augmentation strategy. One can use policy search method on data augmentation [74] with our RényiCL similar to that of [75]. Also, we think that the proposed scheme based on Rényi divergence could be useful for other tasks, e.g., information bottleneck [76]. Lastly, many works focused on the diverse perspectives of contrastive learning [45, 77, 78], where those approaches could be complementary to our work. We leave them for future works.

Negative societal impacts. Since contrastive learning often requires long epochs of training, it raises environmental concerns, e.g. carbon generation. Nevertheless, RényiCL is shown to be effective even under a smaller number of epochs, compared to existing schemes.

Acknowledgement

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00959, Few-shot Learning of Causal Inference in Vision and Language for Decision Making).

References

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

[2] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

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

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[7] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*, 2018.

[8] John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*, 2020.

[9] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

[10] Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3875–3879. IEEE, 2021.

[11] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6964–6974, 2021.

[12] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Active contrastive learning of audio-visual video representations. *arXiv preprint arXiv:2009.09805*, 2020.

[13] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12475–12486, 2021.

[14] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[15] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.

[16] Alfréd Rényi. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, volume 4, pages 547–562. University of California Press, 1961.

[17] Jeremiah Birrell, Paul Dupuis, Markos A Katsoulakis, Luc Rey-Bellet, and Jie Wang. Variational representations and neural network estimation of rényi divergences. *SIAM Journal on Mathematics of Data Science*, 3(4):1093–1116, 2021.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[19] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9588–9597, 2021.

[20] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

[21] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33:21798–21809, 2020.

[22] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2785–2795, 2022.

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

[24] Jiaming Song and Stefano Ermon. Multi-label contrastive predictive coding. *arXiv preprint arXiv:2007.09852*, 2020.

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

[26] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[27] Yao-Hung Hubert Tsai, Shaojie Bai, Louis-Philippe Morency, and Ruslan Salakhutdinov. A note on connecting barlow twins with negative-sample-free contrastive learning. *arXiv preprint arXiv:2104.13712*, 2021.

[28] Sherjil Ozair, Corey Lynch, Yoshua Bengio, Aaron Van den Oord, Sergey Levine, and Pierre Sermanet. Wasserstein dependency measure for representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[29] Yao-Hung Hubert Tsai, Martin Q Ma, Muqiao Yang, Han Zhao, Louis-Philippe Morency, and Ruslan Salakhutdinov. Self-supervised representation learning with relative predictive coding. *arXiv preprint arXiv:2103.11275*, 2021.

[30] Ching-Yao Chuang, R Devon Hjelm, Xin Wang, Vibhav Vineet, Neel Joshi, Antonio Torralba, Stefanie Jegelka, and Yale Song. Robust contrastive learning against noisy views. *arXiv preprint arXiv:2201.04309*, 2022.

[31] Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, 2014.

[32] Tom Minka et al. Divergence measures and message passing. Technical report, Citeseer, 2005.

[33] Yingzhen Li and Richard E Turner. Rényi divergence variational inference. *Advances in neural information processing systems*, 29, 2016.

[34] Liqun Chen, Chenyang Tao, Ruiyi Zhang, Ricardo Henao, and Lawrence Carin Duke. Variational inference and model selection with generalized evidence bounds. In *International conference on machine learning*, pages 893–902. PMLR, 2018.

[35] Chenyang Tao, Liqun Chen, Ricardo Henao, Jianfeng Feng, and Lawrence Carin Duke. Chi-square generative adversarial network. In *International conference on machine learning*, pages 4887–4896. PMLR, 2018.

[36] Lantao Yu, Jiaming Song, Yang Song, and Stefano Ermon. Pseudo-spherical contrastive divergence. *Advances in Neural Information Processing Systems*, 34:22348–22362, 2021.

[37] Josip Djolonga, Mario Lucic, Marco Cuturi, Olivier Bachem, Olivier Bousquet, and Sylvain Gelly. Precision-recall curves using information divergence frontiers. In *International Conference on Artificial Intelligence and Statistics*, pages 2550–2559. PMLR, 2020.

[38] Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on Pure and Applied Mathematics*, 28 (1):1–47, 1975.

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

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

[41] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019.

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

[43] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[44] Art B Owen. Monte carlo theory, methods and examples. 2013.

[45] Qing Guo, Junya Chen, Dong Wang, Yuewei Yang, Xinwei Deng, Lawrence Carin, Fan Li, and Chenyang Tao. Tight mutual information estimation with contrastive fenchel-legendre optimization. *arXiv preprint arXiv:2107.01131*, 2021.

[46] Yang Song and Diederik P Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

[47] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[48] Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John Canny, and Ian Fischer. Compressive visual representations. *Advances in Neural Information Processing Systems*, 34, 2021.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[50] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[51] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[52] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

[53] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[54] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[55] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[56] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[57] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[58] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[59] Xiao Wang and Guo-Jun Qi. Contrastive learning with stronger augmentations. *arXiv preprint arXiv:2104.07713*, 2021.

[60] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[61] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[62] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018.

[63] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

[64] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[65] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[66] Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):1–9, 2014.

[67] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[68] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. In *ICLR*, 2021.

[69] Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

[70] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

[71] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[72] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *arXiv preprint arXiv:1908.01000*, 2019.

[73] Yuning You, Tianlong Chen, Yang Shen, and Zhangyang Wang. Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR, 2021.

[74] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[75] Colorado J Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer. Selfaugment: Automatic augmentation policies for self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2674–2683, June 2021.

[76] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.

[77] Junya Chen, Zhe Gan, Xuan Li, Qing Guo, Liqun Chen, Shuyang Gao, Tagyoung Chung, Yi Xu, Belinda Zeng, Wenlian Lu, et al. Simpler, faster, stronger: Breaking the log-k curse on contrastive learners with flatnce. *arXiv preprint arXiv:2107.01152*, 2021.

[78] Jeff Z HaoChen, Colin Wei, Adrien Gaidon, and Tengyu Ma. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *Advances in Neural Information Processing Systems*, 34:5000–5011, 2021.

[79] Lang Liu, Krishna Pillutla, Sean Welleck, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. Divergence frontiers for generative models: Sample complexity, quantization effects, and frontier integrals. *Advances in Neural Information Processing Systems*, 34, 2021.

[80] Paul Rubenstein, Olivier Bousquet, Josip Djolonga, Carlos Riquelme, and Ilya O Tolstikhin. Practical and consistent estimation of f-divergences. *Advances in Neural Information Processing Systems*, 32, 2019.

[81] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[82] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[83] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[84] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009.

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

[86] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

A Proofs

A.1 Proof of Theorem 4.1

Proof. Without loss of generality, assume $e^{f^*} = dP/dQ$. Then we have

$$\mathrm{Var}_Q[e^{f^*}] = \mathbb{E}_Q\left[\left(\frac{dP}{dQ}\right)^2\right] - \left(\mathbb{E}_Q\left[\frac{dP}{dQ}\right]\right)^2 = \mathbb{E}_P\left[\frac{dP}{dQ}\right] - 1, \tag{9}$$

since $\mathbb{E}_Q[dP/dQ] = 1$. Then the variance of n i.i.d random variable gives us

$$\mathrm{Var}_Q\left[\mathbb{E}_{Q_n}[e^{f^*}]\right] = \frac{\mathrm{Var}[e^{f^*}]}{n} \to 0 \text{ as } n \to \infty$$

For a sequence of random variable X_n defined with respect to distribution Q, and assume we have $\lim_{n\to\infty} X_n = \mathbb{E}X$, then the following comes from the delta method:

$$\lim_{n\to\infty} n\mathrm{Var}_Q[f(X_n)] = (f'(\mathbb{E}[X]))^2\mathrm{Var}_Q[X]. \tag{10}$$

Thus, by applying $f(t) = t^\gamma$ and $\mathbb{E}_Q[dP/dQ] = 1$ gives us

$$\begin{aligned}\lim_{n\to\infty} n\mathrm{Var}_Q[\mathbb{E}_{Q_n}[e^{\gamma f^*}]] &= \gamma^2\mathrm{Var}_Q[e^{f^*}] \\ &= \gamma^2(\mathbb{E}_P[dP/dQ] - 1) \\ &\geq e^{\gamma^2\mathbb{E}_P[\log(dP/dQ)]} - \gamma^2 \\ &= e^{\gamma^2 D_{\mathrm{KL}}(P\|Q)} - \gamma^2,\end{aligned} \tag{11}$$

where the last two equations are from Jensen's inequality and the definition of KL divergence. Now, by applying $f(x) = \log x$ in (10) , we have

$$\begin{aligned}\lim_{n\to\infty} n\mathrm{Var}_Q[\log\mathbb{E}_{Q_n}[e^{\gamma f^*}]] &= \lim_{n\to\infty}\frac{n\mathrm{Var}_Q[\mathbb{E}_{Q_n}[e^{\gamma f^*}]]}{(\mathbb{E}_{Q_n}[e^{\gamma f^*}])^2} \\ &= \lim_{n\to\infty}\frac{n\mathrm{Var}_Q[\mathbb{E}_{Q_n}[e^{\gamma f^*}]]}{e^{2\gamma(\gamma-1)R_\gamma(P\|Q)}} \\ &\geq \frac{e^{\gamma^2 D_{\mathrm{KL}}(P\|Q)} - \gamma^2}{e^{2\gamma(\gamma-1)R_\gamma(P\|Q)}}\end{aligned}$$

from the definition of Rényi divergence and (11). Thus, we have the following:

$$\lim_{n\to\infty} n\mathrm{Var}[\mathcal{I}^{m,n}_{\text{Rényi}}[e^{f^*}]] \geq \lim_{n\to\infty} n\mathrm{Var}[\log\mathbb{E}_{Q_n}[e^{f^*}]] \geq \frac{e^{\gamma^2 D_{\mathrm{KL}}(P\|Q)} - \gamma^2}{e^{2\gamma(\gamma-1)R_\gamma(P\|Q)}}.$$

A.2 Proof of Theorem 4.2

In this section, we state and define the regularity assumptions to derive the asymptotic upper bounds for the variance of α-skew KL divergence in Theorem 3.2. We first review the definition of f-divergence. Let $f : (0,\infty) \to \mathbb{R}$ be a convex function with $f(1) = 0$. Let P and Q be distributions with respect to a base measure dx on domain $\mathcal{X}$, and assume $P \ll Q$. Then the f-divergence generated by f is defined by

$$D_f(P\|Q) \coloneqq \mathbb{E}_Q\left[f\left(\frac{dP}{dQ}\right)\right].$$

Note that for any $c \in \mathbb{R}$, $D_{f_c}(P\|Q) = D_f(P\|Q)$, where $f_c(t) = f(t) + c(t-1)$. Hence, w.l.o.g, we assume $f(t) \geq 0$ for all $t \in (0,\infty)$. The conjugate of f is a function $f^* : (0,\infty) \to [0,\infty)$ defined by $f^*(t) = tf(1/t)$, where $f^*(0) = \lim_{t\to 0^+} f^*(t)$ for convenience. The conjugate f^* of convex function f is also convex. Also, one can see that $f^*(1) = 0$ and $f^*(t) \geq 0$ for all $t \in (0,\infty)$, thus it induces another divergence D_{f^*}. The conjugate divergence D_{f^*} satisfies $D_{f^*}(P\|Q) = D_f(Q\|P)$.

The KL divergence is a f-divergence generated by $f(t) = t\log t - t + 1$, and the α-skew KL divergence is a f-divergence generated by

$$f^{(\alpha)}(t) = t\log\left(\frac{t}{\alpha t + 1 - \alpha}\right) - (1-\alpha)(t-1).$$

From [79], we state following regularity assumptions on the functions f and f^*.

Assumption A.1. *The generator f is twice continuously differentiable with $f'(1) = 0$. Moreover*

(A1) *We have $C_0 := f(0) < \infty$ and $C_0^* := f^*(0) < \infty$.*

(A2) *There exist constants $C_1, C_1^* < \infty$ such that for any $t \in (0, 1)$, we have,*

$$|f'(t)| \le C_1 \max\{1, \log(1/t)\}, \quad \textit{and} \quad |(f^*)'(t)| \le C_1^* \max\{1, \log(1/t)\}.$$

(A3) *There exist constants $C_2, C_2^* < \infty$ such that for every $t \in (0, \infty)$, we have,*

$$\frac{t}{2} f''(t) \le C_2, \quad \textit{and} \quad \frac{t}{2}(f^*)''(t) \le C_2^* .$$

We refer authors [79] for the detailed discussion on Assumption A.1. Then one can observe that KL divergence does not satisfy Assumption A.1, because KL divergence can be unbounded. On the other hand, the α-skew KL divergence satisfies Assumption A.1 from following proposition.

Proposition A.1 ([79])**.** *The α-skew KL divergence generated by $f^{(\alpha)}$ satisfies Assumption A.1 with*

$$C_0 = 1 - \alpha, \quad C_0^* = \log \frac{1}{\alpha} - 1 + \alpha, \quad C_1 = 1, \quad C_1^* = \frac{(1-\alpha)^2}{\alpha}, \quad C_2 = \frac{1}{2}, \quad C_2^* = \frac{1-\alpha}{8\alpha} .$$

For the general f-divergences which satisfy Assumption A.1, the following concentration bound holds.

Proposition A.2 ([79])**.** *Assume f satisfies Assumption A.1, and let P and Q be two distributions with $P \ll Q$. Let P_m be m i.i.d samples from P and Q_n be n i.i.d samples from Q. Then the f-divergence D_f satisfies following:*

$$\mathbb{P}[\,|D_f(P_m \| Q_n) - \mathbb{E}[D_f(P_m \| Q_n)]| > \varepsilon\,] \le 2 \exp\left(- \frac{\varepsilon^2}{\frac{2}{m}(C_1 \log m + c_1)^2 + \frac{2}{n}(C_1^* \log n + c_2)^2} \right)$$

where $c_1 = \max\{C_0^*, C_2\}$ and $c_2 = \max\{C_0, C_2^*\}$.

Thus, the following lemma derives a concentration bound for the α-skew KL divergence by plugging the constants in Proposition A.1 to Proposition A.2.

Lemma A.1. *For $\alpha < \frac{1}{8}$, the following holds:*

$$\mathbb{P}[\,|D_{KL}^{(\alpha)}(P_m \| Q_n) - \mathbb{E}[D_{KL}^{(\alpha)}(P_m \| Q_n)]| > \varepsilon\,] \le 2 \exp\left(- \frac{\varepsilon^2}{\frac{2}{m} \log^2(\alpha m) + \frac{2}{\alpha^2 n} \log^2(e^{1/8} n)} \right)$$

Proof. Note that $C_0^* = \log(1/\alpha) - 1 + \alpha \ge C_2 = 1/2$, and $C_0 = 1 - \alpha \le C_2^* = \frac{1-\alpha}{8\alpha}$ for $\alpha < \frac{1}{8}$. Then the concentration bound follows from Proposition A.2. □

Lastly, we present following upper bound on the bias of empirical estimator of KL divergence:

Proposition A.3 ([80])**.** *Suppose $P \ll Q$, and Var$[dP/dQ] < \infty$. Then we have*

$$|\mathbb{E}[D_{KL}(P_m \| Q_n)] - D_{KL}(P \| Q)| \le \frac{\chi^2(P \| Q)}{\min\{n, m\}}.$$

From proposition A.3, we have

$$|\mathbb{E}[D_{\mathrm{KL}}^{(\alpha)}(P_m \| Q_n)] - D_{\mathrm{KL}}^{(\alpha)}(P \| Q)| \le \frac{\chi^2(P \| \alpha P + (1-\alpha) Q)}{\min\{n, m\}},$$

where $\chi^2(P \| \alpha P + (1-\alpha)Q) = \int \frac{d^2 P}{\alpha dP + (1-\alpha) dQ} \le \int \frac{1}{\alpha} dP = \frac{1}{\alpha}$, or $\int \frac{d^2 P}{\alpha dP + (1-\alpha) dQ} \le \frac{1}{1-\alpha} \int \frac{d^2 P}{dQ} = \frac{\chi^2(P \| Q)}{1-\alpha}$. Therefore, we have

$$|\mathbb{E}[D_{\mathrm{KL}}^{(\alpha)}(P_m \| Q_n)] - D_{\mathrm{KL}}^{(\alpha)}(P \| Q)| \le \frac{c(\alpha)}{\min\{n, m\}}, \quad \text{for} \quad c(\alpha) := \min\left\{ \frac{1}{\alpha}, \frac{\chi^2(P \| Q)}{1-\alpha} \right\}.$$

Now we present the proof of Theorem 4.2 in the main paper.

Proof of Theorem 4.2. Define

$$B_1 := |D_{\mathrm{KL}}^{(\alpha)}(P_m \| Q_n) - \mathbb{E}[D_{\mathrm{KL}}^{(\alpha)}(P_m \| Q_n)]|$$
$$B_2 := |\mathbb{E}[D_{\mathrm{KL}}^{(\alpha)}(P_m \| Q_n)] - D_{\mathrm{KL}}^{(\alpha)}(P \| Q)|$$
$$B_3 := |\mathcal{I}_{\mathrm{KL}}^{(\alpha)}(\widehat{f}) - D_{\mathrm{KL}}^{(\alpha)}(P_m \| Q_n)|.$$

From Proposition A.3, we have $B_2 \leq \frac{c_2}{\min\{n,m\}}$ for some constant $c(\alpha) > 0$, and from the assumption, we have $B_3 \leq \varepsilon_f$. By using triangle inequality twice, we have

$$\begin{aligned} B_1 &\geq |D_{\mathrm{KL}}^{(\alpha)}(P_m \| Q_n) - D_{\mathrm{KL}}^{(\alpha)}(P \| Q)| - B_2 \\ &\geq |D_{\mathrm{KL}}^{(\alpha)}(P \| Q) - \mathcal{I}_{\mathrm{KL}}^{(\alpha)}(\widehat{f})| - B_3 - B_2. \end{aligned}$$

Therefore, it follows that

$$\begin{aligned} \mathbb{P}[|D_{\mathrm{KL}}^{(\alpha)}(P \| Q) - \mathcal{I}_{\mathrm{KL}}^{(\alpha)}(\widehat{f})| > \varepsilon] &\leq \mathbb{P}[B_1 + B_2 + B_3 > \varepsilon] \\ &\leq \mathbb{P}\left[B_1 > \varepsilon - \varepsilon_f - \frac{c(\alpha)}{\min\{n,m\}}\right] \\ &\leq 2\exp\left(-\frac{(\varepsilon - \varepsilon_f - c(\alpha)/\min\{n,m\})^2}{\frac{2}{m}\log^2(\alpha m) + \frac{2}{\alpha^2 n}\log^2(e^{1/8}n)}\right), \end{aligned}$$

where the last inequality comes from Proposition A.2. Since the following holds for any random variable X,

$$\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2] = \int_0^\infty \mathbb{P}[|X - \mathbb{E}X|^2 > t]dt = \int_0^\infty \mathbb{P}[|X - \mathbb{E}X| > \sqrt{t}]dt,$$

we have

$$\mathrm{Var}_{P,Q}\left[\mathcal{I}_{\mathrm{KL}}^{(\alpha)}(\widehat{f})\right] \leq \int_0^\infty 2\exp\left(-\frac{(\sqrt{t} - \varepsilon_f - c(\alpha)/\min\{n,m\})^2}{\frac{2}{m}\log^2(\alpha m) + \frac{2}{\alpha^2 n}\log^2(e^{1/8}n)}\right) dt,$$

which proves our result.

A.3 Proof sketch for Rényi divergence

In this section, we provide a proof sketch for the upper bound for the variance of RMLCPC objective. We use similar proof technique that we used in previous section. Recall that the Rényi Divergence is related to γ-divergence[2] D_γ, a f-divergence generated by $f^{(\gamma)}(t) = \frac{t^\gamma - 1}{\gamma(\gamma-1)}$. The KL divergence is recovered from γ-divergence if we let $\gamma \to 1$. Then we have following equation for Rényi divergence and γ-divergence:

$$R_\gamma(P \| Q) = \frac{1}{\gamma(\gamma-1)}\log(\gamma(\gamma-1)D_\gamma(P \| Q) + 1).$$

One can observe that $f(x) = \frac{1}{\gamma(\gamma-1)}\log(\gamma(\gamma-1)x + 1)$ is of 1-Lipschitz function for any $\gamma \neq 0, 1$. Thus, if we can derive concentration bound for α-skew γ-divergence, we can derive concentration bound for Rényi divergence.

Here, we provide an example when $\gamma = 2$. Remark that when $\gamma = 2$, the γ-divergence is equivalent to χ^2-divergence, which is generated by $f(t) = (t-1)^2$. Then the α-skew χ^2-divergence is generated by following generators:

$$f^{(\alpha)}(t) = \frac{(t-1)^2}{\alpha t + 1 - \alpha} = (f^{(1-\alpha)})^*(t).$$

[2] we use γ for consistency, in general, it is called α-divergence.

Then from [79], the α-skew χ^2-divergence satisfies Assumption A.1 with

$$C_0 = \frac{1}{1-\alpha}, \quad C_0^* = \frac{1}{\alpha}, \quad C_1 = \frac{2}{(1-\alpha^2}, \quad C_1^* = \frac{2}{\alpha^2}, \quad C_2 = \frac{4}{27\alpha(1-\alpha)^2}, \quad C_2^* = \frac{4}{27\alpha^2(1-\alpha)}.$$

Then by using Proposition A.2, for $\alpha < \min\{\frac{2}{3\sqrt{3}}, 1 - \frac{2}{3\sqrt{3}}\}$, we have

$$\mathbb{P}[|D_{\chi^2}^{(\alpha)}(P_m\|Q_n) - \mathbb{E}[D_{\chi^2}^{(\alpha)}(P_m\|Q_n)]| > \varepsilon] \leq 2\exp\left(-\frac{\varepsilon^2}{\frac{c_1}{n}\log^2(c_2 n) + \frac{c_3}{\alpha^2 m}\log^2(c_4 m)}\right),$$

for some constant $c_1, c_2, c_3, c_4 > 0$, then from the Lipschitz continuity, we have

$$\mathbb{P}[|R_2^{(\alpha)}(P_m\|Q_n) - \mathbb{E}[R_2^{(\alpha)}(P_m\|Q_n)]| > \varepsilon] \leq 2\exp\left(-\frac{\varepsilon^2}{\frac{c_1}{n}\log^2(c_2 n) + \frac{c_3}{\alpha^2 m}\log^2(c_4 m)}\right),$$

Thus, by deriving the asymptotic bound for the bias $|\mathbb{E}[R_2^{(\alpha)}(P_m\|Q_n)] - R_2^{(\alpha)}(P\|Q)|$, and if the Rényi variational objective has sufficiently small error, we can derive upper bound for the variance of RMLCPC objective.

B Details for Experiments

B.1 Implementation

As explained in Section 3.3, we derive the full equivalent form of MLCPC and RMLCPC with non-zero α. Recall that the α-MLCPC objective with neural network f_θ is given by

$$\mathcal{I}_{\texttt{MLCPC}}^{(\alpha)}(f_\theta) = \mathbb{E}_{v,v^+}[f_\theta(v,v^+)] - \log\left(\alpha\mathbb{E}_{v,v^+}[e^{f_\theta(v,v^+)}] + (1-\alpha)\mathbb{E}_{v,v^-}[e^{f_\theta(v,v^-)}]\right), \tag{12}$$

then the gradient of (12) with respect to parameter θ is given by

$$\begin{aligned}
&\nabla_\theta \mathcal{I}_{\texttt{MLCPC}}^{(\alpha)}(f_\theta) \\
&= \mathbb{E}_{v,v^+}[\nabla_\theta f_\theta(v,v^+)] - \frac{\alpha\mathbb{E}_{v,v^+}[e^{f_\theta(v,v^+)}\nabla_\theta f_\theta(v,v^+)] + (1-\alpha)\mathbb{E}_{v,v^-}[e^{f_\theta(v,v^-)}\nabla_\theta f_\theta(v,v^-)]}{\alpha\mathbb{E}_{v,v^+}[e^{f_\theta(v,v^+)}] + (1-\alpha)\mathbb{E}_{v,v^-}[e^{f_\theta(v,v^-)}]} \\
&= \mathbb{E}_{v,v^+}[\nabla_\theta f_\theta(v,v^+)] - \left(\mathbb{E}_{\texttt{sg}(q_\theta(v,v^+))}[\nabla_\theta f_\theta(v,v^+)] + \mathbb{E}_{\texttt{sg}(q_\theta(v,v^-))}[\nabla_\theta f_\theta(v,v^-)]\right),
\end{aligned}$$

where

$$\begin{aligned}
q_\theta(v,v^+) &\propto \frac{\alpha p(v,v^+)e^{f_\theta(v,v^+)}}{\alpha\mathbb{E}_{v,v^+}[e^{f_\theta(v,v^+)}] + (1-\alpha)\mathbb{E}_{v,v^-}[e^{f_\theta(v,v^-)}]} \\
q_\theta(v,v^-) &\propto \frac{(1-\alpha)p(v)p(v^-)e^{f_\theta(v,v^-)}}{\alpha\mathbb{E}_{v,v^+}[e^{f_\theta(v,v^+)}] + (1-\alpha)\mathbb{E}_{v,v^-}[e^{f_\theta(v,v^-)}]},
\end{aligned}$$

are importance weights for positive and negative pairs in the second term of (12). For the generalized (α,γ)-RMLCPC, recall that the RMLCPC with neural network f_θ is given by

$$\mathcal{I}_{\texttt{RMLCPC}}^{(\alpha,\gamma)}(f_\theta) = \frac{1}{\gamma-1}\log\mathbb{E}_{v,v^+}[e^{(\gamma-1)f_\theta(v,v^+)}] - \frac{1}{\gamma}\log\left(\alpha\mathbb{E}_{v,v^+}[e^{\gamma f_\theta(v,v^+)}] + (1-\alpha)\mathbb{E}_{v,v^-}[e^{\gamma f_\theta(v,v^-)}]\right). \tag{13}$$

Then the gradient of (13) with respect to θ is given by

$$\begin{aligned}
&\nabla_\theta \mathcal{I}_{\texttt{RMLCPC}}^{(\alpha,\gamma)}(f_\theta) \\
&= \mathbb{E}_{q_\theta^{(1)}(v,v^+)}[\nabla_\theta f_\theta(v,v^+)] - \frac{\alpha\mathbb{E}_{v,v^+}[e^{\gamma f_\theta(v,v^+)}\nabla_\theta f_\theta(v,v^+)] + (1-\alpha)\mathbb{E}_{v,v^-}[e^{\gamma f_\theta(v,v^-)}\nabla_\theta f_\theta(v,v^-)]}{\alpha\mathbb{E}_{v,v^+}[e^{\gamma f_\theta(v,v^+)}] + (1-\alpha)\mathbb{E}_{v,v^-}[e^{\gamma f_\theta(v,v^-)}]} \\
&= \mathbb{E}_{\texttt{sg}(q_\theta^{(1)}(v,v^+))}[\nabla_\theta f_\theta(v,v^+)] - \left(\mathbb{E}_{\texttt{sg}(q_\theta^{(2)}(v,v^+))}[\nabla_\theta f_\theta(v,v^+)] + \mathbb{E}_{\texttt{sg}(q_\theta^{(2)}(v,v^-))}[\nabla_\theta f_\theta(v,v^-)]\right),
\end{aligned}$$

Table 7: Ablation on the effect of α when training with harder data augmentation.

α^{-1}	1024	4096	16384	65536
Base Aug.	79.0	79.3	78.6	78.4
Hard Aug.	81.1	81.3	**81.6**	81.1
Gap	+2.1	+2.0	**+3.0**	+2.7

where

$$q_\theta^{(1)}(v, v^+) \propto p(v, v^+) e^{(\gamma-1) f_\theta(v, v^+)}$$
$$q_\theta^{(2)}(v, v^+) \propto \frac{\alpha p(v, v^+) e^{\gamma f_\theta(v, v^+)}}{\alpha \mathbb{E}_{v, v^+}[e^{\gamma f_\theta(v, v^+)}] + (1-\alpha)\mathbb{E}_{v, v^-}[e^{\gamma f_\theta(v, v^-)}]}$$
$$q_\theta^{(2)}(v, v^-) \propto \frac{(1-\alpha) p(v) p(v^-) e^{\gamma f_\theta(v, v^-)}}{\alpha \mathbb{E}_{v, v^+}[e^{\gamma f_\theta(v, v^+)}] + (1-\alpha)\mathbb{E}_{v, v^-}[e^{\gamma f_\theta(v, v^-)}]},$$

are importance weights for positive and negative pairs for the first and second term in (13). As discussed in Section 3.3, regardless of the insertion of parameter α, MLCPC and RMLCPC explicitly conduct hard-negative sampling, and especially RMLCPC, conducts easy-positive sampling.

The PyTorch style pseudo-code for our implementation on RényiCL is demonstrated in Algorithm 1. In our default implementation, we use multi-crop data augmentation [43] with two global views and multiple local views. In Algorithm 1, we implement the negative of (α, γ)-RMLCPC, and α-MLCPC which is equivalent to the case when $\gamma \to 1$ for RMLCPC, for the contrastive losses. Note that our pseudo-code for contrastive objectives are based on the approach that uses importance weights.

Effect of α in RényiCL. One can observe that the gradient of positive pairs in the second term is multiplied by α, thus the gradient of positive pairs is affected by α and negative pairs. However, since we use a small value of α, the gradient of the second term can be ignored in practice. To verify our claim, we experimented with different values of α on both basic and hard data augmentations:

In Table 7, one can observe that as α becomes smaller, the gap between using hard augmentation and the base augmentation becomes larger. Thus, there is a tradeoff in the choice of α: the α should be large enough to evade large variance (Thm 3.2), but small α is preferred to have the effect of easy positive sampling.

B.2 Further ablation studies

In this section, we provide further ablation studies to validate the effectiveness of RényiCL.

Fair comparison with other self-supervised methods. To see the effectiveness of RényiCL, we conduct additional experiments for fair comparison with other self-supervised methods. MoCo v3 [6] is a state-of-the-art method in contrastive self-supervised learning. For fair comparison of RényiCL with MoCo v3, we apply harder data augmentations such as RandAugment [50] and multi-crops [43] on MoCo v3. We first reimplemented MoCo v3 with smaller batch size (original 4096 to 1024) and in our setting, we achieved 69.6%, which is better than their original reports (68.9%). Then we use harder data augmentations RandAugment (RA), RandomErasing (RE), and multi-crops (MC) when training MoCo v3. Second, we compare RényiCL with DINO [81]. Note that DINO uses default multi-crops, therefore we further applied RandAugment (RA) and RandomErasing (RE) for a fair comparison. For each model, we train for 100 epochs. The results are shown in Table 8. One can observe that while MoCo v3 and DINO benefit from using harder data augmentation, RenyiCL shows the best performance when using harder data augmentation. Also, compared to the Base data augmentation, RényiCL attains the most gain by using harder data augmentation.

Robustness to data augmentations. To see the robustness of RényiCL on the choice of data augmentation, we conduct additional ablation studies on the robustness of RényiCL on various data augmentation schemes. We first show that RényiCL is more robust to the noisy data augmentation, as similar to that in [30], we apply Noisy Crop, which additionally perform RandomResizedCrop of

Algorithm 1 RényiCL: PyTorch-like Pseudocode

```
# f_q: base encoder: backbone + proj mlp + pred mlp
# f_k: momentum encoder: backbone + proj mlp
# m: momentum coefficient
# tau: temperature
# aug_g : global view data augmentation
# aug_l : local view data augmentation

for x in loader: # load a minibatch x with N samples
    x1, x2 = aug_g(x), aug_g(x) # two global views
    k1, k2 = f_k(x1), f_k(x2) # keys: [N, D] each
    q1, q2 = [], [] # list of queries
    q1.append(f_q(x1)), q2.append(f_q(x2)) # queries: [N, D] each
    for i in range(n_crops):
        x_l = aug_l(x)
        q1.append(f_q(x_l)), q2.append(f_q(x_l)) # pass local views only to base encoder

    pos1, pos2, neg1, neg2 = [], [], [], []
    for j in range(n_crops + 1):
        pos, neg = extract_pos_neg(q1[j], k2) # extract pos, neg from q1 and k2
        pos1.append(pos)
        neg1.append(neg)

        pos, neg = extract_pos_neg(q2[j], k1) # extract pos, neg from q2 and k1
        pos2.append(pos)
        neg2.append(neg)
    pos1, pos2, neg1, neg2 = pos1.cat(), pos2.cat(), neg1.cat(), neg2.cat()
    loss = ctr_loss(pos1, neg1) + ctr_loss(pos2, neg2) # symmetrized

    update(f_q) # optimizer update: f_q
    f_k = m*f_k + (1-m)*f_q # momentum update: f_k

# extract positive pairs and negative pairs
def extract_pos_and_neg(q, k):
    # N : number of samples in minibatch
    logits = mm(q, k.t()) / tau # matrix multiplication
    pos = logits.diag() # size N
    neg = logits.flatten()[1:].view(N-1, N+1)[:,:-1].reshape(N,N-1) # size N x (N-1)
    return pos, neg

# contrastive losses using importance weights
def ctr_loss(pos, neg, alpha, gamma):
    pos_d = pos.detach(), neg_d = neg.detach() # no gradient for importance weights
    if gamma == 1: # MLCPC
        loss_1 = -1 * pos.mean()
        iw2_p, iw2_n = pos_d.exp(), neg_d.exp() # importance weights
        loss_2 = alpha*(pos*iw2_p.exp()).mean()+(1-alpha)*(neg*iw2_n.exp()).mean()
        loss_2 /= alpha*iw2_p.mean() + (1-alpha)*iw2_n.mean()
        loss = loss_1 + loss_2
    elif gamma > 1: # RMLCPC
        iw1 = ((gamma - 1)*pos_d).exp() # importance weight for first term
        loss_1 = (pos*iw1).mean() / iw1.mean()
        iw2_p, iw2_n = (gamma*pos_d).exp(), (gamma*neg_d).exp() # importance weights
        loss_2 = alpha*(pos*iw2_p).mean() + (1-alpha)*(neg*iw2_n).mean()
        loss_2 /= alpha*iw2_p.mean() + (1-alpha)*iw2_n.mean()
        loss = loss_1 + loss_2
    return loss
```

size 0.2. Then the generated views might contain only nuisance information (e.g. background), thus it can hurt the generalization of contrastive learning. We train each model with CPC, MLCPC, and RMLCPC objectives and data augmentations with and without Noisy Crop. We experimented on CIFAR-10, CIFAR-100, and ImageNet-100 datasets and the results are shown in Table 9. One can observe that while the performance of using CPC and MLCPC degrades as we apply the noisy crop, RMLCPC conversely improves the performance. Hence, RMLCPC is robust in the choice of data augmentation.

Second, we experiment when there is limited data augmentation. For each CIFAR-10, CIFAR-100, and STL-10 dataset, we remove data augmentations such as color jittering and grayscale and only apply RandomResizedCrop and HorizontalFlip. The results are in Table 10. Note that RMLCPC achieves the best performance among CPC and MLCPC, when we do not apply data augmentation. Therefore we show that the RMLCPC objective is not only robust to the hard data augmentation but is also robust when there is limited data augmentation.

Table 8: Fair comparison with other self-supervised learning methods by using same hard data augmentation. All models are run by us with 100 training epochs.

Method	DINO [81]	MoCo v3 [6]	RényiCL
Base	70.9	69.6	69.4
Hard	72.5	73.5	**74.3**
Gain	+1.6	+3.9	**+4.9**

Table 9: Ablation on the robustness of RényiCL to Noisy Crop.

Method	CIFAR-10	CIFAR-100	ImageNet-100
CPC	91.7	65.4	79.7
+NC	90.6(-1.1)	64.9(-0.5)	79.2(-0.5)
MLCPC	91.9	65.6	79.5
+NC	90.7(-1.2)	65.0(-0.6)	79.4(-0.1)
RMLCPC	90.7	64.5	78.9
+NC	**91.6(+0.9)**	**67.6(+3.1)**	**80.5(+1.6)**

Table 10: Ablation on the robustness of RényiCL when learning with limited data augmentation.

Method	CIFAR-10	CIFAR-100	STL-10
CPC	63.2	31.8	52.8
MLCPC	63.3	32.9	53.1
RMLCPC	**67.1**	**41.2**	**56.4**

B.3 ImageNet experiments

Data augmentation. For the baseline, we follow the good practice in existing works [2, 3, 47, 6], which includes random resized cropping, horizontal flipping, color jittering [2], grayscale conversion [2], Gaussian blurring [3], and solarization [47]. Given the baseline data augmentation, we add the following data augmentation methods:

- RandAugment [50]: a strong data augmentation strategy that includes 14 image-specific transformations such as AutoContrast, Posterize, etc. We use default settings that were used in [50] (2 operations at each iteration and a magnitude scale of 10).
- RandomErasing [65]: a simple data augmentation strategy that randomly masks a box in an image with a random value. The RandomErasing was applied with probability 0.2.
- Multi-crop [43]: a multi-view data augmentation strategy for self-supervised learning, we use two global crops of size 224 and 2 or 6 local crops of size 96. For global crops, we scale the image with $(0.2, 1.0)$ and for the local crops, we scale the image with $(0.05, 0.2)$, which is slightly different from that in [43].

For the implementation of RandAugment and RandomErasing, we use the implementation in [82]. Note that when using Multi-crop, we only use RandAugment on the global views, and we do not use RandomErasing, which slightly degrades performance when used for Multi-crops. Otherwise, if we do not use Multi-crop, we use both RandAugment and RandomErasing.

Model. We use ResNet-50 [49] for backbone. For our base encoder, we further use a projection head [3], and a prediction head [47]. The momentum encoder is updated by the moving average of the base encoder, except that we do not use the prediction head for momentum encoder [6]. The momentum coefficient is initialized by 0.99 and gradually increased by the half-cycle cosine schedule to 1. We use 2-layer MLP with dimensions 2048-4096-256 for the projection head. We attach the batch-normalization (BN) layer and ReLU activation layer at the end of each fully-connected layer, except for the last one, to which we only attach the BN layer. For the prediction head, we use 2-layer MLP with dimensions of 256-4096-256, and we attach BN and ReLU at the end of each fully connected layer except the last one. Then the critic is implemented by the temperature-scaled cosine similarity: $f(x, y; \tau) = x^\top y/(\tau\|x\|_2\|y\|_2)$, with $\tau = 0.5$. For RMLCPC objective, we use $\gamma = 2.0$, and $\alpha = 1/65536$.

Optimization. We use LARS [83] optimizer with a batch size of 512, weight decay of 1.5×10^{-6}, and momentum of 0.9 for all experiments. We use root scaling for learning rate [3], which was

Table 11: Dataset information for the transfer learning tasks. For FC100, CUB200, Plant Disease, we perform few-shot learning, otherwise, we perform linear evaluation.

Dataset	# of classes	Training	Validation	Test	Metric
CIFAR10 [51]	10	45000	5000	10000	Top-1 accuracy
CIFAR100 [51]	100	45000	5000	10000	Top-1 accuracy
Food [52]	101	68175	7575	25250	Top-1 accuracy
MIT67 [84]	67	4690	670	1340	Top-1 accuracy
Pets [60]	37	2940	740	3669	Mean per-class accuracy
Flowers [53]	102	1020	1020	6149	Mean per-class accuracy
Caltech101 [61]	101	2525	505	5647	Mean Per-class accuracy
Cars [54]	196	6494	1650	8041	Top-1 accuracy
Aircraft [55]	100	3334	3333	3333	Mean Per-class accuracy
DTD (split 1) [56]	47	1880	1880	1880	Top-1 accuracy
SUN397 (split 1) [57]	397	15880	3970	19850	Top-1 accuracy
VOC2007 [58]	20	4952	2501	2510	Mean average precision
FC100 [62]	20	-	-	12000	Average accuracy
CUB200 [50]	200	-	-	11780	Average accuracy
Plant Disease [63]	38	-	-	54305	Average accuracy

effective in our framework. When training for 100 epochs, we use the base learning rate of 0.15, and for training for 200 epochs, we use the base learning rate of 0.075. The learning rate is gradually annealed by a half-cycle cosine schedule. We also used automatic mixed precision for scalability.

Linear evaluation. For ImageNet linear evaluation accuracy, we report the validation accuracy of the linear classifier trained on the top of frozen features. We use random resized cropping of size 224 and random horizontal flipping for training data augmentation, and resizing to the size of 256 and center-crop of 224 for validation data augmentation. We use cross-entropy loss for training, where we train for 90 epochs with SGD optimizer, batch size of 1024, and base learning rate of 0.1, where we use linear learning rate scaling and learning rate follows cosine annealing schedule. We do not use weight decay.

Semi-supervised learning. For semi-supervised learning on ImageNet, we follow the protocol as in [3, 47, 43]. We add a linear classifier on the top of frozen representation, then train both backbone and classifier using either 1% or 10% of the ImageNet training data. For a fair comparison, we use the splits proposed in [3]. We use the same data augmentation and loss function as in linear evaluation. As same as previous works, we use different learning rates for each backbone and classification head. For the 1% setting, we use a base learning rate of 0.01 for backbone and 0.2 for classification head. For the 10% setting, we use a base learning rate of 0.08 for the backbone and 0.03 for the classification head. The learning rate is decayed with a factor of 0.2 at 16 and 20 epochs. We do not use weight decay or other regularization technique. We train for 20 epochs with SGD optimizer, batch size of 512 for both 1% and 10% settings, and do not use weight decay.

Transfer learning. For transfer learning, we train a classifier on the top of frozen representations as done in many previous works [3, 47]. In Table 11, we list the information of the datasets we used; the number of classes, number of samples for each train/val/test split, and evaluation metric. We use the same data augmentation that we used for linear evaluation on the ImageNet dataset. For optimization, we ℓ_2-regularized L-BFGS, where the regularization parameter is selected from a range of 45 logarithmically spaced values from 10^{-6} to 10^{5} using the validation split. After the best hyperparameter is selected, we train the linear classifier using both training and validation splits and report the test accuracy using the metric instructed in Table 3. The maximum number of iterations in L-BFGS is 5000 and we use the previous solution as an initial point, i.e., a warm start, for the next step. For few-shot learning experiments, we perform logistic regression on the top of frozen representations and use $N \times K$ support samples without fine-tuning and data augmentation in a N-way K-shot episode.

Table 12: Comparison of contrastive self-supervised methods with harder augmentations by downstream few-shot classification accuracy (%) over 2000 episodes on FC100, CUB200, and Plant disease datasets. (N, K) denotes N-way K-shot classification. † denotes the usage of multi-crop data augmentation. **Bold** entries denote the best performance.

	FC100		CUB200		Plant Disease	
Method	(5,1)	(5,5)	(5,1)	(5,5)	(5,1)	(5,5)
InfoMin [14]	31.80	45.09	53.81	72.20	66.11	84.12
CLSA [59]	34.46	49.16	50.83	67.93	67.39	86.15
CLSA† [59]	41.09	58.38	52.87	70.92	71.57	88.94
RényiCL	36.31	53.39	**59.73**	82.12	80.68	94.29
RényiCL†	**42.10**	**60.80**	58.25	**82.38**	**83.40**	**95.71**

Table 13: Comparison of contrastive self-supervised methods with harder augmentations by downstream linear evaluation accuracy on various datasets. IN denotes ImageNet validation accuracy (%), and † denotes the use of multi-crop data augmentation. **Bold** entries denote the best performance.

Method	IN	CIFAR10	CIFAR100	Food101	Pets	MIT67	Flowers	Caltech101	Cars	Aircraft	DTD	SUN397
InfoMin [14]	73.0	92.8	75.8	73.8	86.4	76.9	91.2	88.5	49.6	50.5	75.0	61.2
CLSA [59]	72.2	94.2	77.4	73.2	85.2	77.1	90.8	91.2	48.7	51.6	75.0	62.0
CLSA† [59]	73.3	93.9	76.9	73.4	86.4	77.7	91.2	90.2	47.4	50.0	75.5	62.9
RényiCL	72.6	93.8	78.8	72.7	88.4	75.8	94.2	**94.0**	62.1	58.8	74.6	62.1
RényiCL†	**75.3**	**94.4**	**78.9**	**77.5**	**89.2**	**81.1**	**96.2**	**94.0**	**66.4**	**61.1**	**76.3**	**65.9**

Extended version of Table 4. We provide additional experimental results in the comparison with other contrastive representation learning methods that use stronger augmentations. The InfoMin [14] used RandAugment [50] and Jigsaw cropping for data augmentation and trained for 800 epochs. The CLSA [59] used both RandAugment [50] and pyramidal multi-crop data augmentation, which generates additional crops of size $192 \times 192, 160 \times 160, 128 \times 128$, and 96×96. For comparison, we use the checkpoints from their official github repositories[3] [4]. In Table 12, we report the few-shot classification accuracies (%) on FC100 [62], CUB200 [50], and Plant Disease [63] datasets, which is the extended version of Table 3 in the main paper. In Table 13, we report the linear evaluation accuracies (%) on various object classification datasets extending the results in Table 3 in the main paper.

B.4 CIFAR experiments

For CIFAR experiments, we use random resized cropping with a size of 32×32, random horizontal flipping, color jittering, and random grayscale conversion with probability 0.2 for the base data augmentation. For harder data augmentation, we apply RandAugment [50], where we use 3 operations at each iteration with a magnitude scale of 5, and RandomErasing [65]. We use modified ResNet-18 [49], where the kernel size of the first convolutional layer is converted from 7×7 to 3×3, and we do not use Max pooling at the penultimate layer [3]. We attach a 2-layer projection head of size 512-2048-128 with ReLU and BN layer attached at each FC layer, except for the last one. We do not use a prediction head, and the temperature is set to be 0.5 for every CIFAR experiment. We use $\gamma = 1.5$ and $\alpha = 1/4096$ for both CIFAR-10 and CIFAR-100 experiments. We train for 500 epochs with the SGD optimizer, learning rate of 0.5 (without linear scaling), weight decay of 5e-4, and momentum of 0.9. For evaluation, we train a linear classifier on the top of frozen features with 100 epochs with an SGD optimizer, learning rate of 0.3, without using weight decay. We only used random resized crop with a size of 32, and random horizontal flipping for both training and evaluation.

B.5 Tabular experiments

For the data augmentation of tabular experiments, we consider simple random masking (RM) noise that was used in [68], and random feature corruption (FC) proposed in [69]. The random masking noise is analogous to RandomErasing [65] that we used in image contrastive learning. The FC data

[3]InfoMin: https://github.com/HobbitLong/PyContrast
[4]CLSA: https://github.com/maple-research-lab/CLSA

Table 14: Full experimental results for CovType and Higgs-100K datasets. We report top-1 accuracy (%) with linear evaluation. Average over 5 runs. **Bold** entries denote the best performance.

	CovType				Higgs-100K			
Method	No Aug.	RM	FC	RM+FC	No Aug.	RM	FC	RM+FC
CPC	71.6± 0.36	69.8± 0.26	74.3± 0.44	73.6± 0.16	64.7± 0.22	71.3± 0.08	64.9± 0.11	71.4± 0.15
MLCPC	71.7± 0.26	70.2± 0.15	74.1± 0.39	74.0± 0.45	64.9± 0.11	71.5± 0.06	65.3± 0.26	71.5± 0.06
RMLCPC (γ=1.1)	72.1± 0.31	70.9± 0.42	**74.9±0.28**	73.8± 0.47	65.1± 0.29	71.8± 0.17	65.2± 0.13	71.9± 0.06
RMLCPC (γ=1.2)	71.9± 0.58	70.8± 0.25	74.5± 0.38	73.7± 0.37	64.5± 0.41	**72.4±0.13**	65.3± 0.44	72.3± 0.10

augmentation randomly selects a subset of attributes on each tabular data and mixes it with the attributes from another data. Note that the random corruption is performed in the whole dataset scale [69]. We refer to [69] for detailed implementation of FC data augmentation. We use a probability of 0.2 for both random masking noise and FC data augmentation. For the backbone, we use a 5-layer MLP with a dimension of 2048-2048-4096-4096-8192, where all layers have a BN layer followed by RELU, except for the last one. We use MaxOut activation for the last layer with 4 sets, and on the top of the backbone, we attach 2-layer MLP with a dimension of 2048-2048-128 as a projection head. For Higgs-100K, we set the $\tau = 0.1$, $\gamma = 1.2$, and $\alpha = 1/4096$. For CovType, we set $\tau = 0.2$, $\gamma = 1.1$, and $\alpha = 1/4096$. We train for 500 epochs with an SGD optimizer of momentum 0.9, base learning rate of 0.3, and the learning rate is scheduled by half-cycle cosine annealing. For evaluation of the CovType experiments, we train a linear classifier on the top of frozen representation for 100 epochs with a learning rate in $\{1, 3, 5\}$ and report the best validation accuracy (%). For evaluation of the Higgs-100K experiments, we use linear regression on the top of frozen representations by pseudo-inverse and report the validation accuracy (%). In Table 14, we provide the full experimental results on CovType and Higgs-100K. One can notice that the RM data augmentation is effective for the Higgs dataset, and FC data augmentation is effective for the CovType dataset.

B.6 Graph experiments

For graph contrastive learning experiments on TUDataset [70], we follow the same experimental setup in [15]. Thus, we adopt the official implementation[5], and experiment over the hyper-parameter $\gamma \in \{1.5, 2.0, 3.0\}$ and choose the best parameter by evaluating on the validation dataset.

C Mutual Information Estimation

C.1 Implementation details

Assume we have B batches of samples $\{x_i\}_{i=1}^B$ from $x_i \sim X$ and B batches of samples $\{y_i\}_{i=1}^B$ from $y_i \sim Y$. Assume $(x_i, y_i) \sim P_{XY}$, i.e. positive pair, and $(x_i, y_j) \sim P_X P_Y$ for $i \neq j$. Then let $f^*(x, y)$ be optimal critic learned by maximizing CPC, MLCPC, or RMLCPC objectives. From the optimality condition, we have

$$f^*(x, y) \propto \log \frac{dP_{XY}(x, y)}{\alpha dP_{XY}(x, y) + (1 - \alpha) dP_X(x) dP_Y(y)} = \log \frac{r(x, y)}{\alpha r(x, y) + 1 - \alpha},$$

where $r(x, y) = \frac{dP_{XY}}{dP_X dP_Y}$ is a true density ratio. Then our idea is to recover r by using f^* and approximate the true mutual information. To do that, we first compute the log-normalization constant Z, i.e. $\frac{e^{f^*(x,y)}}{Z} = \frac{r(x,y)}{\alpha r(x,y)+1-\alpha}$. We use Monte-Carlo [44] methods by computing empirical mean as following:

$$\hat{Z} = \frac{\alpha}{B} \sum_{i=1}^{B} e^{f^*(x_i, y_i)} + \frac{1 - \alpha}{B(B - 1)} \sum_{i=1}^{B} \sum_{j \neq i} e^{f^*(x_i, y_j)}.$$

Then we have following approximate density ratio $\hat{r}$:

$$\hat{r}(x, y) = \frac{(1 - \alpha) e^{f^*(x,y)}}{\hat{Z} - \alpha e^{f^*(x,y)}}.$$

[5] https://github.com/Shen-Lab/GraphCL/tree/master/unsupervised_TU

Since the mutual information is defined by the empirical mean of log-density ratio, we have

$$\hat{\mathcal{I}}(X;Y) = \frac{1}{B}\sum_{i=1}^{B} \log \hat{r}(x_i, y_i)$$

C.2 MI estimation between correlated Gaussians

Setup. We follow the general procedure in [39, 24]. We sample a pair of vectors $(x, y) \in \mathbb{R}^{d\times d}$ from random variable $X \times Y$, where X and Y are unit normal distribution, and are correlated with correlation coefficient $\rho \in (0, 1)$. Then the true mutual information (MI) between X and Y can be computed by $\mathcal{I}(X;Y) = -\frac{d}{2}\log\left(1-\frac{\rho}{2}\right)$. In our experiments, we set the initial mutual information to be 2, and we increase the MI by $\times 2$ for every 4K iterations. We consider joint critic, which concatenates the inputs x, y and then passes through a 2-layer MLP with a dimension of $d - 256 - 1$, and the output of the fully connected layers are followed by the ReLU activation layer. For optimization, we use the Adam optimizer with learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, batch size of 128.

Results. In Figure , we depict the curve of training curves and MI estimation that we proposed in Section 3.4. We consider α-CPC, α-MLCPC, (α, γ)-RMLCPC with $\gamma = 2.0$. Remark that when $\alpha = 0.0$, the MI estimators exhibit large variance as we proved in Theorem 3.1. But for large enough $\alpha > 0$, the training objectives tend to have low variance, and thus we have a low variance MI estimator.

From Theorem 4.1, since there are total 128×127 negative pairs, we need to select $\alpha \propto 1/128$. Therefore, we divide the α by 128 for each experiment in Figure 2. One can observe that even though the optimal α varies across the objectives, choosing α around 1/128 gives a fairly good estimation of MI.

Note that the skew divergence is not only applicable to DV objectives, but also for the NWJ [85] objective, which is another variational lower bound of KL divergence defined as following:

$$\mathcal{I}_{\texttt{NWJ}}(f) \coloneqq \mathbb{E}_{P_{XY}}[f(x,y)] - \mathbb{E}_{P_X P_Y}[e^{f(x,y)-1}],$$

and we have $\mathcal{I}(X;Y) = D_{\texttt{KL}}(P_{XY}\|P_X P_Y) = \sup_f \mathcal{I}_{\texttt{NWJ}}(f)$ [85]. Note that the NWJ objective is proven to have large variance as the true MI is too large [41]. However, we show that by introducing the NWJ objective for α-skew KL divergence, one can lower the variance of the training objective, and perform MI estimation as we explained in Section 3.4. We refer the NWJ variational lower bound of α-skew KL divergence as α-NWJ, and is defined by following:

$$\mathcal{I}^{(\alpha)}_{\texttt{NWJ}}(f) \coloneqq \mathbb{E}_{P_{XY}}[f(x,y)] - \alpha\mathbb{E}_{P_{XY}}[e^{f(x,y)-1}] - (1-\alpha)\mathbb{E}_{P_X P_Y}[e^{f(x,y)-1}],$$

and we have $D^{(\alpha)}_{\texttt{KL}}(P_{XY}\|P_X P_Y) = \sup_f \mathcal{I}^{(\alpha)}_{\texttt{NWJ}}(f)$. Lastly, in Figure , we show that one can perform stable MI estimation through α-NWJ objective by choosing appropriate α.

C.3 Mutual Information between the views and InfoMin principle.

Implementation. We present the pseudocode for estimation of MI between the views in Algorithm 2. In Figure , we compute the MI for every iteration, and report the mean value of MI for the last epoch of training.

InfoMin principle and RényiCL. Following [14], we plot the mutual information versus linear evaluation accuracy (%) by training with different data augmentations and contrastive objectives. In Figure 1, we observe that for harder data augmentations, the RMLCPC induces lower MI than that of CPC and MLCPC, while showing better downstream performances. This is because of the effect of easy positive and hard negative sampling explained in Section 4.3. Note that InfoMin [14] principle argues that the optimal views must share sufficient and minimal information about the downstream tasks. Here, one can observe that RényiCL intrinsically follows the InfoMin principle as it extracts minimal and sufficient information between the views. Therefore, RényiCL intrinsically follows InfoMin principle [14] that it extracts minimal and sufficient information between the views. Here, we establish the orthogonal relationship between the InfoMin principle [14] and RényiCL: InfoMin principle aims to find views that share minimal and sufficient information, while RényiCL can extract minimal and sufficient information from given views.

Algorithm 2 Pseudocode for MI estimation between the views

```
# aug: data augmentation that generates views
# x: batch of samples
# f: critic that takes a pair of inputs
# extract_pos_and_neg: a function that extracts positive and negative pairs (Alg. 1)

def MI_estimation(x, f, aug, alpha, gamma):
    x1, x2 = aug(x), aug(x) # two views
    pos, neg = extract_pos_and_neg(f(x1, x2)) # extract positive and negative pairs
    Z = alpha*pos.exp() + (1-alpha)*neg.exp().mean(dim=1) # normalization constant
    r_alpha = pos.exp() / Z
    r_true = (1-alpha)*r_alpha / (1-alpha*r_alpha)
    return r_true.log().mean()
```

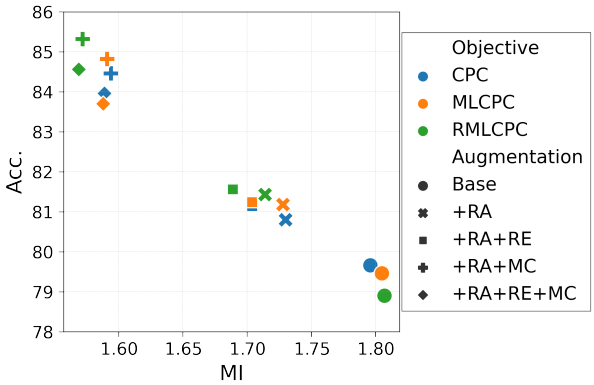


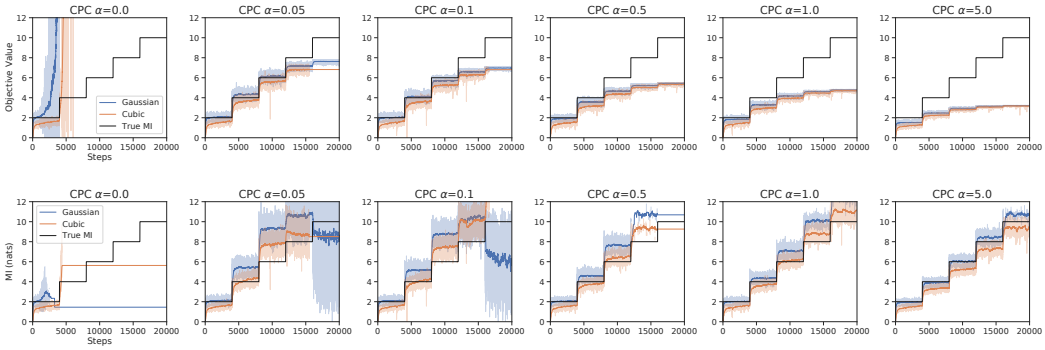
Figure 1: MI between the views and validation accuracy on ImageNet-100. RA: RandAugment [50], RE: RandomErasing [65], MC: Multi-crops [43].

D Hyperparameter Sensitivity Analysis

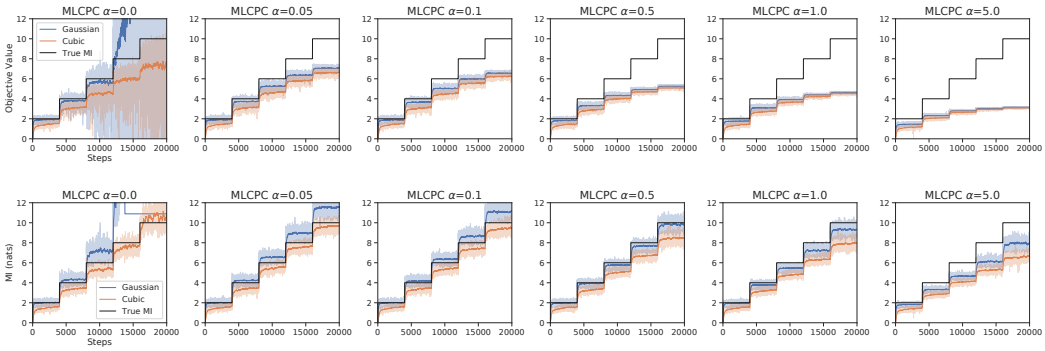
The temperature is a hyperparameter that greatly affects the performance of contrastive learning [3, 86]. Here, we show that RMLCPC objective is robust to the choice of temperature hyperparameter. We conduct simple ablation studies on CIFAR-10 and CIFAR-100 with temperature $\tau = 0.5, 1.0$ and $\gamma = 1.5, 2.0$. We use the same experimental setup described in B.4 except the value of temperature. For comparison, we use same data augmentation setup; base data augmentation and hard augmentation using RandAugment and RandomErasing. In Table 15, we report the linear evaluation accuracy of each setting trained on CIFAR-10 and CIFAR-100 datasets. When using large temperature, i.e., $\tau = 1.0$, CPC and MLCPC degrades their performance when using harder data augmentation. On the other hand, RMLCPC consistently achieves better performance when using harder augmentations on both $\tau = 0.5$ and $\tau = 1.0$. Especially for CIFAR-100 dataset, using RMLCPC objective with $\tau = 1.0$ shows similar performance with using MLCPC objective with $\tau = 0.5$. Thus, our RényiCL does not only provide empirical gain, but also it could be a good starting point to search for effective hyperparameter due to their robustness to temperature.

Table 15: Sensitivity of CPC, MLCPC, RMLCPC on temperature. We report the linear evaluation accuracy (%) on CIFAR-10 and CIFAR-100 datasets. **Bold** entries denote the best performance.

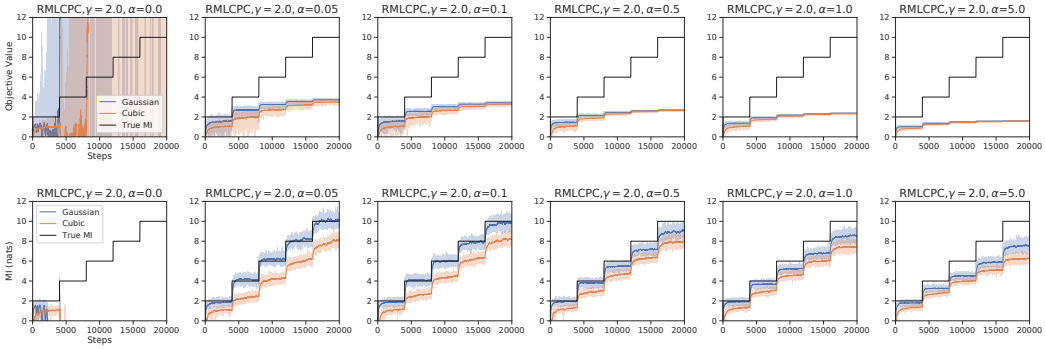
	CIFAR-10				CIFAR-100			
	$\tau = 1.0$		$\tau = 0.5$		$\tau = 1.0$		$\tau = 0.5$	
Method	Base	Hard	Base	Hard	Base	Hard	Base	Hard
CPC	91.7	90.9	91.7	91.9	63.0	62.2	65.4	67.1
MLCPC	91.6	90.8	91.9	92.1	62.8	61.9	65.6	66.6
RMLCPC ($\gamma = 1.5$)	91.6	91.7	91.4	**92.5**	64.2	64.2	65.4	68.0
RMLCPC ($\gamma = 2.0$)	91.3	92.2	90.7	**92.5**	64.5	66.2	64.5	**68.5**



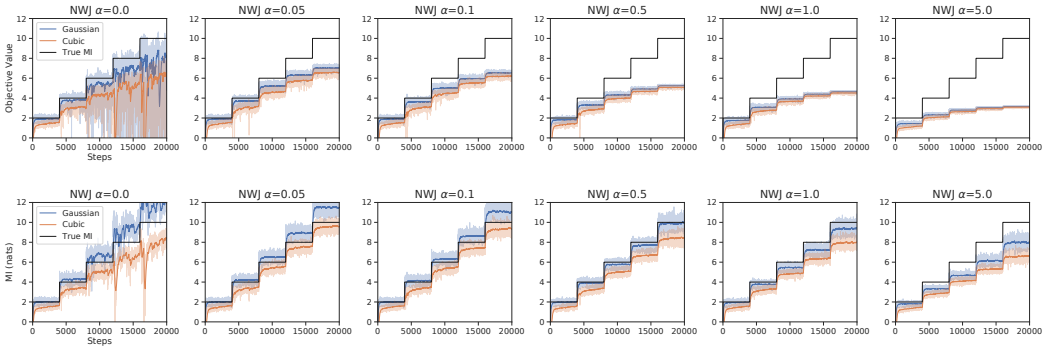
(a) Top: α-CPC objective, Bottom: MI estimation from α-CPC



(b) Top: α-MLCPC objective, Bottom: MI estimation from α-MLCPC



(c) Top: (α, γ)-RMLCPC objective, Bottom: MI estimation from (α, γ)-RMLCPC



(d) Top: α-NWJ, Bottom: MI estimation from α-NWJ

Figure 2: The plots for Gaussian MI estimation. We plot the values of training objectives, i.e., the negative of each objective, and the estimated MI value. For RMLCPC objective, we use $\gamma = 2.0$. Note that we divide the α by the batch size of 128.