f-Mutual Information Contrastive Learning

Guojun Zhang[*]
University of Waterloo
Huawei Noah's Ark Lab
guojun.zhang@uwaterloo.ca

Yiwei Lu[*]
University of Waterloo
Vector Institute
y485lu@uwaterloo.ca

Sun Sun
National Research Council Canada
Sun.Sun@nrc.cnrc.gc.ca

Hongyu Guo
National Research Council Canada
hongyu.guo@nrc.cnrc.gc.ca

Yaoliang Yu
University of Waterloo
Vector Institute
yaoliang.yu@uwaterloo.ca

Abstract

Self-supervised contrastive learning is an emerging field due to its power in providing good data representations. Such learning paradigm widely adopts the InfoNCE loss, which is closely connected with maximizing the mutual information. In this work, we propose the f-Mutual Information Contrastive Learning framework (f-MICL), which directly maximizes the f-divergence-based generalization of mutual information. We theoretically prove that, with mild assumptions, our f-MICL naturally attains the *alignment* for positive pairs and the *uniformity* for data representations, the two main factors for the success of contrastive learning. We further provide theoretical guidance on designing the similarity function and choosing the effective f-divergences for f-MICL. Using several benchmark tasks, we empirically verify that our novel method outperforms or performs on par with state-of-the-art strategies.

1 Introduction

Contrastive learning has attracted a surge of attention recently due to its success in learning informative representation for image recognition, natural language understanding, and reinforcement learning [Chen et al., 2020, He et al., 2020, Logeswaran and Lee, 2018, Srinivas et al., 2020]. Such learning paradigm is fully unsupervised by encouraging the contrastiveness between similar and dissimilar sample pairs. Specifically, the feature embeddings of similar sample pairs are expected to be close while those of dissimilar sample pairs are expected to be far apart. To attain this goal, a softmax cross-entropy loss, a.k.a. InfoNCE, has been widely used [Wu et al., 2018, van den Oord et al., 2018, Chen et al., 2020, Hénaff et al., 2020, He et al., 2020], which aims to maximize the probability of picking a similar sample pair among a batch of sample pairs.

InfoNCE can be interpreted as a lower bound of the mutual information (MI) between two views of data samples [van den Oord et al., 2018, Bachman et al., 2019, Tian et al., 2020a, Tschannen et al., 2020]. This explanation is consistent with the well-known "InfoMax principle" [Linsker, 1988]. Nevertheless, it has been shown that maximizing a tighter bound on the MI can result in worse

[*] Equal contribution

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

representations [Tschannen et al., 2020]; and reducing the MI between views while only keeping task-relevant information can improve the downstream performance [Tian et al., 2020b]. These observations suggest that maximizing the MI may be insufficient in contrastive learning and thus a better objective design is required.

To attain this goal, we propose a novel contrastive learning framework, coined as f-MICL. In a nutshell, leveraging the fact that MI can be formulated as the Kullback–Leibler (KL) divergence between the joint distribution and the product of the marginal distributions, we replace the KL divergence with the general f-divergence family [Ali and Silvey, 1966, Csiszár, 1967]. Doing so, we obtain a generalization of mutual information, called *f-mutual information* [f-MI, Csiszár, 1967]. Notably, by maximizing a lower bound of f-mutual information we naturally decompose the objective function into two terms, which correspond to the properties of the *alignment* and the *uniformity*. Such characterization has been revealed in Wang and Isola [2020, Theorem 1] for the InfoNCE loss. Compared with Wang and Isola [2020], our result applies to a wide range of the f-divergence family, and it does not rely on the limit of an infinite number of dissimilar samples. This allows us to explore the space of f-MI and improve the performance of InfoNCE-based contrastive learning.

The similarity function is crucial for the evaluation of the contrastiveness of similar and dissimilar sample pairs. Commonly used similarity functions include the cosine similarity [Chen et al., 2020, He et al., 2020], the bilinear functions [van den Oord et al., 2018, Tian et al., 2020a, Hénaff et al., 2020], and the neural network based scores [Hjelm et al., 2018]. While most aforementioned similarity functions for contrastive learning are heuristic and pre-designed, in this work, we provide a principled way to design the similarity function. By assuming that the joint feature distribution of two similar samples is proportional to a Gaussian kernel, we derive an optimal similarity function for practical use, which resembles the well-known radial basis functions [Powell, 1987]. With the optimization of our f-mutual information objective the positive pairs are aligned with each other and the data representations are uniformly distributed.

2 f-Mutual Information Contrastive Learning

In this work we propose the f-mutual information framework for contrastive learning. First recall the f-mutual information (f-MI) between a pair of random variables X and Y:

Definition 1 (**f-mutual information, Csiszár 1967**). *Consider a pair of random variables (X, Y) with density function $p(x, y)$. The f-mutual information I_f between X and Y is defined as*

$$I_f(X;Y) := D_f\left(p(x,y)\|p(x)p(y)\right) = \int f\left(\tfrac{p(x,y)}{p(x)p(y)}\right) p(x)p(y) \cdot \mathrm{d}\lambda(x,y), \tag{1}$$

where $f : \mathbb{R}_+ \to \mathbb{R}$ is (closed) convex with $f(1) = 0$, and recall that $p(x)$ and $p(y)$ are the marginal densities of $p(x, y)$, whereas λ is a dominating measure (e.g. Lebesgue).

Common choices of f can be found in Table 2 (Appendix A). It is well-known that f-mutual information is non-negative and symmetric. Nguyen et al. [2010] derived a variational method by maximizing the dual problem: $I_f(X;Y) \geq \sup_{T\in\mathcal{T}} i_f(X;Y) := \mathbb{E}_{(x,y)\sim p_{\text{pos}}}[T(x,y)] - \mathbb{E}_{(x,y)\sim p_{\texttt{data}}\otimes p_{\texttt{data}}}[f^*(T(x,y))]$, where $f^*(t) := \sup_{x\in\mathbb{R}_+}(xt - f(x))$ is the (monotone) Fenchel conjugate of f, and is always *monotonically increasing*. Here $\mathcal{T}$ is a class of functions $T : \text{supp}(p_{\texttt{data}}) \times \text{supp}(p_{\texttt{data}}) \to \text{dom } f^*$. Following Chen et al. [2020], we design the structure of function T as: $T(x,y) := k(g(x), g(y))$, where $\|g(x)\| = 1$ for any sample x. The function g produces a d-dimensional normalized feature encoding on the hypersphere $\mathbb{S}^{d-1}$ and k is a similarity function that measures the similarity between two embeddings $g(x)$ and $g(y)$. With the above interpretation, we can rewrite our objective of f-mutual information:

$$\sup_{g\in\mathcal{G},k\in\mathcal{K}} i_f(X;Y) := \mathbb{E}_{(x,y)\sim p_{\text{pos}}}[k(g(x),g(y))] - \mathbb{E}_{(x,y)\sim p_{\texttt{data}}\otimes p_{\texttt{data}}}[f^*(k(g(x),g(y)))], \tag{2}$$

where $\mathcal{G}$ and $\mathcal{K}$ are the function classes of the feature encoder g and the similarity function k. We can treat the first term as the similarity score between *positive pairs* in the feature space, and the second term as the similarity score between two random samples, a.k.a. *negative pairs*, in the feature space. As f^* is monotonically increasing, maximizing f-MI is equivalent to simultaneously maximizing the similarity between positive pairs and minimizing the similarity between negative pairs.

Algorithm 1: f-mutual information contrastive learning (f-MICL)

```
Input: batch size $N$, function $f$, weighting parameter $\alpha$, constant $\mu$ (in $G_\sigma$), variance $\sigma^2$,
       optimizer
for each batch $\{z_i\}_{i=1}^N$ do
    forall $k \in [1, N]$ do
        randomly sample two augmentation functions $t_1, t_2$
        $y_k \leftarrow t_1(z_k)$, $x_k \leftarrow t_2(z_k)$
    compute $i_f = \frac{1}{N}\sum_{i=1}^N \left[f' \circ G_\sigma(\|x_i^g - y_i^g\|^2)\right] - \frac{\alpha}{N(N-1)}\sum_{i \neq j} f^* \circ f' \circ G_\sigma(\|x_i^g - x_j^g\|^2)$
    update $g$ by taking a step to maximizing $i_f$ using the optimizer
```

2.1 Optimized similarity function and implementation

We now study how to search for the optimal similarity function k. For the ease of notations, from now on we define $x^g := g(x)$ and $y^g := g(y)$. Suppose $(x, y) \sim p_{\mathtt{pos}}$, then we denote $p_{\mathtt{pos}}^g$ as the distribution of (x^g, y^g), and $p_{\mathtt{data}}^g$ as the marginal feature distribution of x^g or y^g. The corresponding density functions are written as $p_g(x^g), p_g(y^g)$ and $p_g(x^g, y^g)$. Recall the following result:

Lemma 2 (*e.g.*, Nguyen et al. 2010, Lemma 1)**.** *Suppose f is differentiable, and the encoder function g is fixed. The similarity function $k^*(x^g, y^g) = f'\left(\frac{p_g(x^g, y^g)}{p_g(x^g)p_g(y^g)}\right)$ maximizes $i_f(X; Y)$ in eq.* (2) *as long as it is contained in the function class $\mathcal{K}$.*

Lemma 2 provides an optimal similarity function, which nevertheless depends on the density functions. To use k^* practically we make the following assumption on the joint density:

Assumption 3. *The joint feature distribution is proportional to a Gaussian kernel, i.e., $p_g(x^g, y^g) \propto G_\sigma(\|x^g - y^g\|^2) := \mu \exp\left(-\frac{\|x^g - y^g\|^2}{2\sigma^2}\right)$, with μ a constant left to be determined.*

Fixing y^g, then $p_g(\cdot, y^g)$ mentioned in Assmp. 3 is known as the *von Mises–Fisher* distribution [von Mises, 1918, Fisher, 1953, Bingham and Mardia, 1975], since x^g and y^g are unit vectors. With Assmp. 3 on the joint density, the resultant marginal feature distribution $p_{\mathtt{data}}^g$ is uniform on the hypersphere $\mathbb{S}^{d-1}$, where d is the dimension of the feature space (see Prop. 7 in App. B). Additionally, for positive pairs the distance in the feature space, $\|x^g - y^g\|$, is more likely to be small. Based on Assmp. 3 we propose:

Theorem 4 (**Gaussian similarity**)**.** *Under Assumption 3 with Gaussian kernels and the same settings as Lemma 2, the optimal similarity function k^* satisfies that for any $x^g, y^g \in \mathbb{S}^{d-1}$: $k^*(x^g, y^g) = f'(CG_\sigma(\|x^g - y^g\|^2))$, where d is the feature dimension and C is an absolute constant.*

For simplicity we will rewrite $k^*(x^g, y^g) = f' \circ G_\sigma(\|x^g - y^g\|^2)$ by absorbing the constant C into G_σ, since we have left some flexibility in Assumption 3. Bringing the optimal k^* in Theorem 4 into our objective eq. (2) we have:

$$\sup_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim p_{\mathtt{pos}}}\left[f' \circ G_\sigma(\|x^g - y^g\|^2)\right] - \mathbb{E}_{(x,y)\sim p_{\mathtt{data}} \otimes p_{\mathtt{data}}}\left[f^* \circ f' \circ G_\sigma(\|x^g - y^g\|^2)\right], \tag{3}$$

where G_σ is defined in Assumption 3. We now use a similar sampling method as in Chen et al. [2020]. Given a batch of N samples we can estimate the objective in eq. (3) as:

$$\widehat{i}_f(X; Y) = \frac{1}{N}\sum_{i=1}^N f' \circ G_\sigma(\|x_i^g - y_i^g\|^2) - \frac{1}{N(N-1)}\sum_{i \neq j} f^* \circ f' \circ G_\sigma(\|x_i^g - x_j^g\|^2), \tag{4}$$

where x_i and y_i are two different kinds of data augmentation of the i-th sample, and x_i and x_j are different samples of the same kind of data augmentation. With the objective in equation 4, we propose our algorithm for contrastive learning in Algorithm 1.

2.2 Alignment and Uniformity

Notably, if we choose the f-divergence to be the KL divergence, the objective in equation 3 becomes:

$$\sup_{g \in \mathcal{G}} -\frac{1}{2\sigma^2}\mathbb{E}_{(x,y)\sim p_{\mathtt{pos}}}\left[\|x^g - y^g\|^2\right] - \mu\, \mathbb{E}_{(x,y)\sim p_{\mathtt{data}} \otimes p_{\mathtt{data}}}\left[\exp\left(-\frac{\|x^g - y^g\|^2}{2\sigma^2}\right)\right], \tag{5}$$

which retrieves the objective of the alignment and uniformity in Wang and Isola [2020]. Specifically, in equation 5 the first expectation is the same as $\mathcal{L}_{\texttt{align}}$, and the second expectation is the same as $\mathcal{L}_{\texttt{uniform}}$ in Wang and Isola [2020] (up to the logarithmic transformation).

Let us now study alignment and uniformity for general f-divergences. Consider the objective in eq. (3). For the first term, since f is convex, the derivative f' is monotonic increasing. Therefore, in the ideal case, maximizing the first term would yield $x^g = y^g$ for all $(x, y) \sim p_{\texttt{pos}}$, *i.e.*, similar sample pairs should have aligned representations. For uniformity, we have the following theorem:

Theorem 5 (**uniformity**). *Suppose that the batch size N satisfies $2 \leq N \leq d + 1$, with d the dimension of the feature space. If the real function $h(t) = f^* \circ f' \circ G_\sigma(t)$ is strictly convex on $[0, 4]$, then all minimizers of the second term of eq. (3),* i.e., *$\sum_{i \neq j} f^* \circ f' \circ G_\sigma(\|x_i^g - x_j^g\|^2)$, satisfy that the feature representations of all samples are distributed uniformly on the unit hypersphere $\mathbb{S}^{d-1}$.*

Here by "distributed uniformly" we mean that the feature vectors form a regular simplex, and thus the distances between all sample pairs are the same. It reflects our intuition that the feature embeddings are evenly distributed. The assumption in Theorem 5 $N \leq d + 1$ is always satisfied in our experiments in §3. For instance, for CIFAR-10 experiments we choose $N = d = 512$. Common f-divergences such as KL, Pearson χ^2 and Jensen–Shannon satisfy Theorem 5 and lead to the property of uniformity. However, this conclusion does not hold for all f-divergences, *e.g.*, Reversed Kullback–Leibler (RKL) (App. A.1). Experimentally, we found that the RKL divergence results in feature collapse (*i.e.*, all feature vectors are the same) and thus poor performance.

3 Experiments

We compare our framework with various frameworks on several popular datasets. To achieve fair comparison, we keep the network architecture and optimization the same, while only changing the objective accordingly in our comparison. See Appendix C for more detailed settings. In particular, our f-MICL gives state-of-the-art performance compared to popular choices of the loss functions, such as InfoNCE [van den Oord et al., 2018, Chen et al., 2020], Uniformity [Wang and Isola, 2020], and RPC [Tsai et al., 2020].

Specifically, our results confirm the following: **(1)** Our f-MICL encourages *alignment* between positive pairs, and encourages dissimilar sample pairs to be equally far apart and thus leads to *uniformity*; **(2)** By replacing the cosine similarity with the Gaussian kernels, the performance is consistently better across a variety of f-divergences in our f-MICL framework.

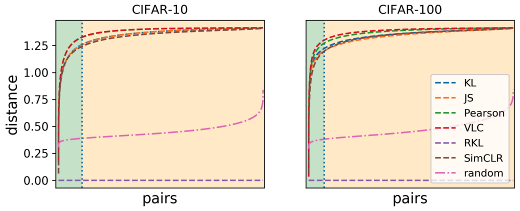

Figure 1: Distances between pairs of normalized features within a batch. **Green region:** similar pairs. **Orange region:** dissimilar pairs. f-MICL gives nearly uniform distances for dissimilar pairs for the f-divergences. For non-satisfying f-divergences such as the RKL, the features collapse to a constant and thus the distances are zero.

3.1 Comparison with benchmarks

We compare with several state-of-the-art benchmarks in Table 1. Note that SimCLR and RPC use the cosine similarity while we use the proposed Gaussian similarity. Our datasets include CIFAR-10, CIFAR-100 [Krizhevsky et al., 2009], STL-10 [Coates et al., 2011], TinyImageNet [Chrabaszcz et al., 2017] and ImageNet [Deng et al., 2009] for image classification. After learning a feature embedding, we evaluate the quality of representation using the test classification accuracies via a linear classifier. We observe from Table 1 that our proposed f-MICL consistently outperforms the benchmarks across all datasets. Specifically, we find that the JS divergence is superior in general, especially in larger datasets.

3.2 Uniformity Test

To check the uniformity of feature vectors (Theorem 5) we plot the pairwise distance $\|x_i^g - x_j^g\|$ of the feature representations within the same batch on CIFAR-10 and CIFAR-100. We compute the distances between the normalized features of every pair from a random batch, and then sort the pairs with the increasing order. From Figure 1 we can see that f-MICL gives nearly uniform distances for dissimilar pairs (orange regions) on both datasets with various proper f-divergences. In contrast, a

Table 1: Test classification accuracy (%) on various datasets with linear evaluation.

Dataset	**Baselines**			***f*-MICL**					
	SimCLR	Uniformity	RPC	KL	JS	Pearson	SH	Tsallis	VLC
CIFAR-10	89.71	90.41	90.39	**90.61**	89.66	89.35	89.52	89.15	89.13
CIFAR-100	62.75	62.51	62.66	63.00	**63.11**	61.69	61.47	60.55	61.19
STL-10	82.97	84.44	82.41	85.33	**85.94**	82.64	82.80	84.79	83.27
TinyImageNet	30.54	41.10	34.93	39.16	**42.88**	38.42	40.87	32.95	38.61
ImageNet	57.66	59.12	56.11	58.91	**61.11**	55.33	52.37	53.11	54.26

random initialized model gives a less uniform distribution for dissimilar pairs. Besides, for f-MICL we observe small pairwise distances for similar pairs (green regions). On the CIFAR-100 dataset we observe that there are less similar pairs compared to CIFAR-10 as there are more classes.

Acknowledgement

We thank the reviewers for constructive comments. We thank NSERC and the Canada CIFAR AI Chairs program for funding support. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute.

References

Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1): 131–142, 1966.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems*, 32: 15535–15545, 2019.

MS Bingham and KV Mardia. Maximum likelihood characterization of the von Mises distribution. In *A Modern Course on Statistical Distributions in Scientific Work*, pages 387–398. Springer, 1975.

Sergiy V Borodachov, Douglas P Hardin, and Edward B Saff. *Discrete energy on rectifiable sets*. Springer, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607, 2020. URL `http://proceedings.mlr.press/v119/chen20j.html`.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of ImageNet as an alternative to the CIFAR datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

Imre Csiszár. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

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.

Ronald Aylmer Fisher. Dispersion on a sphere. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 217(1130):295–305, 1953.

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.

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.

Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. Technical report.

Lucien Le Cam. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media, 2012.

Ralph Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. In *International Conference on Learning Representations*, 2018.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations*, 2017.

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.

M. J. D. Powell. *Radial Basis Functions for Multivariable Interpolation: A Review*, page 143–167. Clarendon Press, USA, 1987. ISBN 0198536127.

Ralph Rockafellar. Characterization of the subdifferentials of convex functions. *Pacific Journal of Mathematics*, 17(3):497–510, 1966.

Igal Sason and Sergio Verdú. f-divergence inequalities. *IEEE Transactions on Information Theory*, 62(11):5973–6006, 2016.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, 2020.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020b. URL `https://proceedings.neurips.cc//paper_files/paper/2020/hash/4c2e5eaae9152079b9e95845750bb9ab-Abstract.html`.

Yao-Hung Hubert Tsai, Han Zhao, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Neural methods for point-wise dependency estimation. *arXiv preprint arXiv:2006.05553*, 2020.

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. In *International Conference on Learning Representations*, 2021.

Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of statistical physics*, 52(1):479–487, 1988.

Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2020.

Jean-Baptiste Hiriart Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms*. Springer-Verlag, 1993.

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

Richard von Mises. Uber die "ganzzahligkeit" der atomgewicht und verwandte fragen. *Phys. Z.*, 19: 490–500, 1918.

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.

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.

Table 2: A summary of common f-divergences. KL: Kullback–Leibler; JS: Jensen–Shannon; SH: Squared Hellinger. For JS, we define $\varphi(u) = -(u+1)\log\frac{1+u}{2} + u\log u$. For Pearson χ^2, we take $f^*(t) = -1$ if $t \leq -2$. For Jeffrey, $\widehat{W} = W + W^{-1}$ and $W(\cdot)$ is the Lambert-W product log function. The Tsallis-α divergence is defined in Tsallis [1988] and we have $\alpha > 1$ for f-divergences. We ignore constant addition $-1/(\alpha-1)$ because it does not change the optimization problem. The Vincze–Le Cam divergence can be found in [p.47, Le Cam, 2012] which is closely related to χ^2 and Hellinger divergences. For the Vincze–Le Cam divergence we require $-3 < t < 1$ and $f^*(t) = -1$ if $t \leq -3$.

Divergence	$f(u)$	$f^*(t)$	$f'(u)$	$f^* \circ f'(u)$
KL	$u\log u$	$\exp(t-1)$	$\log u + 1$	u
Reverse KL	$-\log u$	$-1-\log(-t)$	$-1/u$	$\log u - 1$
JS	$\varphi(u)$	$-\log(2-e^t)$	$\log 2 + \log\frac{u}{1+u}$	$-\log 2 + \log(1+u)$
Pearson χ^2	$(u-1)^2$	$t^2/4 + t$	$2(u-1)$	$u^2 - 1$
SH	$(\sqrt{u}-1)^2$	$\frac{t}{1-t}$	$1 - u^{-1/2}$	$u^{1/2} - 1$
Neyman χ^2	$\frac{(1-u)^2}{u}$	$2 - 2\sqrt{1-t}$	$1 - u^{-2}$	$2 - 2u^{-1}$
Jeffrey	$(u-1)\log u$	$\widehat{W}(e^{1-t}) + t - 2$	$1 - u^{-1} + \log u$	$\widehat{W}(e^{1/u}/u) + \log u - \frac{1+u}{u}$
Tsallis α	$u^\alpha/(\alpha-1)$	$((\alpha-1)t/\alpha)^{\alpha/(\alpha-1)}$	$\frac{\alpha u^{\alpha-1}}{\alpha-1}$	u^α
Vincze–Le Cam	$\frac{(u-1)^2}{u+1}$	$4 - t - 4\sqrt{1-t}$	$\frac{(u-1)(u+3)}{(u+1)^2}$	$3 - \frac{4}{u+1}$

A Additional theoretical results

In this appendix we provide additional theoretical results, including additional f-divergences and the theory for weighting parameters.

A.1 f-divergences

We give examples of f-divergences in Table 2.. A detailed description of f-divergences can be found in e.g. Sason and Verdú [2016].

A.2 Weighting parameters

In Algorithm 1 we added a weighting parameter α to balance the alignment and uniformity. We prove that even after adding this parameter we are still maximizing the f-mutual information, although with respect to a different f.

Proposition 6 (**weighting parameter**). *Given $\alpha > 0$ and a closed convex function $f : \mathbb{R}_+ \to \mathbb{R}$ such that $f(1) = 0$, define $f_\alpha : \alpha\,\mathrm{dom}\, f \to \mathbb{R}$ with $f_\alpha(x) = \alpha f(x/\alpha) - \alpha f(1/\alpha)$ for any $x \in$ dom f. Then I_{f_α} is still a valid f-mutual information (see Definition 1). Besides, by replacing f with f_α in equation 3 we have the following optimization problem:*

$$\sup_{g\in\mathcal{G}} \mathbb{E}_{(x,y)\sim p_{\rm pos}}\left[f'\left(G_\sigma(\|x^g - y^g\|^2)/\alpha\right)\right] - \alpha\mathbb{E}_{(x,y)\sim p_{\rm data}\otimes p_{\rm data}}\left[f^*\circ f'\left(G_\sigma(\|x^g - y^g\|^2)/\alpha\right)\right],$$

where $G_\sigma(\|x^g - y^g\|^2) = \mu\exp\left(-\frac{\|x^g - y^g\|^2}{2\sigma^2}\right)$ is the Gaussian kernel.

Note that $\alpha\,\mathrm{dom}\, f$ means the scalar multiplication of a set which is applied element-wisely. According to Definition 1, f_α is also a valid f-divergence. This proposition tells us that rescaling the second term with factor α is equivalent to changing the function f to another convex function f_α. The transformation from f to $\alpha f(x/\alpha)$ is also known as right scalar multiplication [e.g. Chapter X, Urruty and Lemaréchal, 1993]. Let us now move on to our proof:

Proof. By definition we know that f_α is convex and closed with $f_\alpha(1) = 0$, and thus I_{f_α} is a valid f-mutual information according to Definition 1. Moreover, we have $f'_\alpha(x) = f'(x/\alpha)$ for any

$x \in \alpha \, \mathrm{dom}\, f$ and

$$\begin{aligned}
f_\alpha^*(t) &= \sup_{x \in \mathrm{dom}\, f_\alpha} xt - f_\alpha(x) \\
&= \sup_{x \in \alpha \mathrm{dom}\, f} xt - \alpha f(x/\alpha) + \alpha f(1/\alpha) \\
&= \sup_{x/\alpha \in \mathrm{dom}\, f} (x/\alpha) \cdot (\alpha t) - \alpha f(x/\alpha) + \alpha f(1/\alpha) \\
&= \alpha \sup_{x/\alpha \in \mathrm{dom}\, f} ((x/\alpha) \cdot t - f(x/\alpha)) + \alpha f(1/\alpha) \\
&= \alpha f^*(t) + \alpha f(1/\alpha), \qquad (6)
\end{aligned}$$

where in the last line we used the definition of $f^*(t)$. Plugging f_α' and f_α^* into equation 3 yields the desired result. □

A.3 Uniform distributions

In the following we prove that under Assumption 3 the marginal feature distribution $p^g_{\mathtt{data}}$ is uniform.

Proposition 7. *Under Assumption 3, the marginal feature distribution $p^g_{\mathtt{data}}$ is uniform on the hypersphere $\mathbb{S}^{d-1}$, with d the dimension of the feature space.*

Proof. Under Assumption 3 we have:

$$p_g(x^g, y^g) = C_0 \exp\left(-\frac{\|x^g - y^g\|^2}{2\sigma^2}\right), \qquad (7)$$

where C_0 is a normalizing constant. Integrating y^g we have the marginal distribution for x^g:

$$p_g(x^g) = \int_{\mathbb{S}^{d-1}} C_0 \exp\left(-\frac{\|x^g - y^g\|^2}{2\sigma^2}\right) dy^g. \qquad (8)$$

It suffices to show that $p_g(x_1^g) = p_g(x_2^g)$ for any $x_1^g, x_2^g \in \mathbb{S}^{d-1}$. Suppose Q is the orthogonal matrix such that:

$$Qx_1^g = x_2^g. \qquad (9)$$

Such a matrix Q always exists and constructing Q is not difficult. For example, assume that $\{x_1^g, x_2^g\}$ span a plane with an orthonormal basis e_1, e_2, and

$$\begin{aligned}
x_1^g &= \cos\theta_1 \cdot e_1 + \sin\theta_1 \cdot e_2, \\
x_2^g &= \cos\theta_2 \cdot e_1 + \sin\theta_2 \cdot e_2, \qquad (10)
\end{aligned}$$

then Q can take the following form:

$$Q = [e_1 \; e_2] \begin{bmatrix} \cos(\theta_2 - \theta_1) & -\sin(\theta_2 - \theta_1) \\ \sin(\theta_2 - \theta_1) & \cos(\theta_2 - \theta_1) \end{bmatrix} \begin{bmatrix} e_1^\top \\ e_2^\top \end{bmatrix} \qquad (11)$$

such that Q is orthogonal and satisfies $Qx_1^g = x_2^g$. Hence we have:

$$\begin{aligned}
p_g(x_1^g) &= \int_{\mathbb{S}^{d-1}} C_0 \exp\left(-\frac{\|x_1^g - y^g\|^2}{2\sigma^2}\right) dy^g \\
&= \int_{\mathbb{S}^{d-1}} C_0 \exp\left(-\frac{\|Q^\top x_2^g - y^g\|^2}{2\sigma^2}\right) dy^g \\
&= \int_{\mathbb{S}^{d-1}} C_0 \exp\left(-\frac{\|Q^\top x_2^g - Q^\top z^g\|^2}{2\sigma^2}\right) d(Q^\top z^g) \\
&= \int_{\mathbb{S}^{d-1}} C_0 \exp\left(-\frac{\|x_2^g - z^g\|^2}{2\sigma^2}\right) dz^g \\
&= p_g(x_2^g), \qquad (12)
\end{aligned}$$

where in the second line we used equation 9; in the third line we made the transformation $y^g = Q^\top z^g$ with z^g a unit vector; in the fourth line we used the fact that applying a orthogonal matrix does not change the norm and that the corresponding Jacobian determinant is one.

In fact, the proof above can be generalized from the Gaussian kernel to any radial basis functions, by replacing the Gaussian kernel with $\varphi(\|x^g - y^g\|^2)$, and repeating the same proof. Here φ can be any function such that the integral $\int_{\mathbb{S}^{d-1}} \varphi(\|x^g - y^g\|^2) dy^g$ is finite. □

B Proofs

Lemma 2 (*e.g.*, Nguyen et al. 2010, Lemma 1)**.** *Suppose f is differentiable, and the encoder function g is fixed. The similarity function $k^*(x^g, y^g) = f'\left(\frac{p_g(x^g,y^g)}{p_g(x^g)p_g(y^g)}\right)$ maximizes $i_f(X;Y)$ in eq.* (2) *as long as it is contained in the function class $\mathcal{K}$.*

Proof. From Definition 1, we are computing the following supremum:

$$\sup_{g,k} \int \left(\frac{p_g(x^g, y^g)}{p_g(x^g)p_g(y^g)} k(x^g, y^g) - f^* \circ k(x^g, y^g) \right) dp^g_{\mathtt{data}} \otimes p^g_{\mathtt{data}}. \tag{13}$$

Suppose k is unconstrained and we fix g. The optimal solution should satisfy:

$$\frac{p_g(x^g, y^g)}{p_g(x^g)p_g(y^g)} \in (\partial f^*)(k^*(x^g, y^g)), \tag{14}$$

almost surely for $(x, y) \sim p_{\mathtt{data}} \otimes p_{\mathtt{data}}$. From (3.11) of Rockafellar [1966] this is equivalent to:

$$k^*(x^g, y^g) \in \partial f \left(\frac{p_g(x^g, y^g)}{p_g(x^g)p_g(y^g)} \right). \tag{15}$$

If f is differentiable, then for any $u \in$ dom f, $\partial f(u) = \{f'(u)\}$ is a singleton. □

Theorem 4 (Gaussian similarity). *Under Assumption 3 with Gaussian kernels and the same settings as Lemma 2, the optimal similarity function k^* satisfies that for any $x^g, y^g \in \mathbb{S}^{d-1}$: $k^*(x^g, y^g) = f'(CG_\sigma(\|x^g - y^g\|^2))$, where d is the feature dimension and C is an absolute constant.*

Proof. Simply combine Proposition 7 with Lemma 2. □

Theorem 5 (uniformity). *Suppose that the batch size N satisfies $2 \leq N \leq d + 1$, with d the dimension of the feature space. If the real function $h(t) = f^* \circ f' \circ G_\sigma(t)$ is strictly convex on $[0, 4]$, then all minimizers of the second term of eq. (3),* i.e., *$\sum_{i \neq j} f^* \circ f' \circ G_\sigma(\|x_i^g - x_j^g\|^2)$, satisfy that the feature representations of all samples are distributed uniformly on the unit hypersphere $\mathbb{S}^{d-1}$.*

Proof. From the definition of h it is clear that h is decreasing since f^* and f' are both monotonically increasing white G_σ is decreasing. Using h we rewrite the second term of equation 4 as

$$\min_{x_1^g, \ldots, x_N^g \in \mathbb{S}^{d-1}} \sum_{i,j} h(\|x_i^g - x_j^g\|^2). \tag{16}$$

When $N \in [2, d + 1]$, there exists a neat characterization of the minimizers, see e.g. Borodachov et al. [2019, Theorem 2.4.1]. We include the proof below for completeness.

Apply Jensen's inequality, we have:

$$\begin{aligned} \frac{1}{N^2} \sum_{i,j} h(\|x_i - x_j\|^2) &\geq h \left(\frac{1}{N^2} \sum_{i,j} \|x_i - x_j\|^2 \right) \\ &= h \left(\frac{1}{N^2} \sum_{i,j} \|x_i - x_j\|^2 \right) \\ &= h \left(\frac{1}{N^2} \sum_{i,j} (2 - 2x_i \cdot x_j) \right) \\ &= h \left(2 \left(1 - \left\| \frac{1}{N} \sum_{i=1}^N x_i \right\|^2 \right) \right) \\ &\geq h\,(2)\,, \end{aligned} \tag{17}$$

where in the first line we used Jensen's inequality; in the third line we used $\|x_i\| = \|x_j\| = 1$ for any $i, j \in [N]$; in the last line we note that $\| \sum_{i=1}^N x_i\| \geq 0$ and h is a decreasing function. When h is strictly convex and decreasing, it is in fact strictly decreasing, and hence the two inequalities above can be attained iff

$$\bar{x} := \frac{1}{N}\sum_i x_i = \mathbf{0}, \quad \text{and } \|x_i - x_j\|^2 \equiv c \text{ for all } i \neq j, \tag{18}$$

namely that $\{x_1, \ldots, x_N\}$ form a regular simplex with its center at the origin. We remark that when h is merely convex, points forming a centered regular simplex may form a strict subset of the minimizers.

To see the necessity of $N \leq d + 1$, let us note that

$$x_i^\top x_j = \begin{cases} 1, & i = j \\ -\frac{1}{N-1}, & i \neq j \end{cases}, \tag{19}$$

since

$$\sum_{ij} \|x_i - x_j\|^2 = 2N^2 = N(N-1)c \implies c = \frac{2N}{N-1} = 2 + \frac{2}{N-1}. \tag{20}$$

Performing simple Gaussian elimination we note that the matrix $X^\top X$ has rank $N - 1$ where $X = [x_1, \ldots, x_N] \in \mathbb{R}^{d\times N}$. Therefore, we must have $N - 1 \leq d$.

Lastly, we need to show when h is a (strictly) convex function, which may not always be true depending on the f-divergences. We give the following characterization (we ignore the constants μ and $2\sigma^2$ as they do not affect convexity):

- h strictly convex: $h_{\text{KL}}(t) = e^{-t}$, $h_{\text{JS}}(t) = \log(1 + e^{-t}) - \log 2$, $h_{\text{Pearson}}(t) = e^{-2t} - 1$, $h_{\text{SH}}(t) = e^{-t/2} - 1$, $h_{\text{Tsallis}}(t) = e^{-\alpha t}$, $h_{\text{VLC}} = 3 - \frac{4}{1+e^{-t}}$;
- h convex but not strictly convex: $h_{\text{RKL}}(t) = -t - 1$ (RKL stands for Reversed Kullback–Leibler, see Appendix A.1);
- h concave: $h_{\text{Neyman}}(t) = 2 - 2e^t$ (Neyman stands for Neyman χ^2, see Appendix A.1).

Only for the last case we do not have the guarantee that the minimizing configurations could form a regular simplex. For RKL, in fact, any configuration that centers at the origin suffices since h is a linear function. □

C Experimental details

We present additional experimental details in this appendix, to further support our experiments in the main paper.

In this paper, we follow the implementations in SimCLR (`https://github.com/sthalles/SimCLR`). We use ResNet [He et al., 2016] as the feature encoder, and we adopt the similar procedure of SimCLR for sampling. Our experimental settings are detailed below:

- Hardware and package: We train on a GPU cluster with `NVIDIA T4` and `P100`. The platform we use is `pytorch`. Specifically, the pairwise summation can be easily implemented using `torch.nn.functional.pdist` from `pytorch`.
- Datasets: the datasets we consider include CIFAR-10, CIFAR-100 [Krizhevsky et al., 2009], STL-10 [Coates et al., 2011], TinyImageNet [Chrabaszcz et al., 2017] and ImageNet [Deng et al., 2009].
- Augmentation method: For each sample in a dataset we create a sample pair, a.k.a. positive pair, using two different augmentation functions. For image samples, we choose the augmentation functions to be the standard ones in contrastive learning, e.g., in Chen et al. [2020] and He et al. [2020]. The augmentation is a composition of random flipping, cropping, color jittering and gray scaling.

Table 3: Detailed experimental settings. `arch`: the neural network architecture used. N: batch size; d: the dimension of the feature representation; `lr`: learning rate; μ: the constant factor in μ; $1/(2\sigma^2)$ and α follow from Algorithm 1; `epoch`: the number of epochs we run.

Dataset	`arch`	N	d	`lr`	μ	$(2\sigma^2)^{-1}$	α	`epoch`
CIFAR-10	ResNet-18	512	512	0.1	1	1	40	800
CIFAR-100	ResNet-18	512	512	0.1	1	1	40	1000
STL-10	ResNet-50	64	512	0.1	1	1	40	800
TinyImageNet	ResNet-50	256	512	0.1	1	1	40	800
ImageNet	ResNet-50	256	512	0.1	1	1	40	100

- Neural architecture: For CIFAR-10 and CIFAR-100 we use ResNet-18 [He et al., 2016]; for STL-10,TinyImageNet and ImageNet we use ResNet-50 [He et al., 2016].
- Batch size and embedding dimension: for experiments in CIFAR-10 and CIFAR-100 we choose batch size 512, and for STL-10 we choose batch size 64 to accommodate one GPU training. Finally, for TinyImageNet and ImageNet, we choose batch size 256. For all the datasets, we choose the embedding dimension to be 512. In all of these cases, our assumption $N \leq d + 1$ in Theorem 5 is satisfied.
- Hyperparameters: in all our experiments we fix the constant factor $\mu = 1$. We find that in practice the weight parameter α often needs to be large , which requires moderate tuning. Note that we also implement RPC [Tsai et al., 2021] in our paper. For all the datasets, we follow Tsai et al. [2021] and choose the relative parameters $\alpha = 1.0, \beta = 0.005$ and $\gamma = 1.0$ for all datasets.
- Optimizer and learning rate scheduler: We use SGD with momentum for optimization and the cosine learning rate scheduler [Loshchilov and Hutter, 2017].
- Evaluation metric: we use linear evaluation to evaluate the performance, based on the learned embeddings.

Table 3 gives common choices of hyperparameters for different datasets. Note that we may need to further finetune α and σ for different f-divergences. See our supplementary code for more details.