Simpler, Faster, Stronger: Breaking The $\log$-K Curse On Contrastive Learners With FlatNCE

Junya Chen[1,†], **Zhe Gan**[2], **Xuan Li**[3], **Qing Guo**[3], **Liqun Chen**[4], **Shuyang Gao**[4], **Wenlian Lu**[5]
Tagyoung Chung[4], **Yi Xu**[4], **Belinda Zeng**[4], **Fan Li**[1], **Lawrence Carin**[6], **Chenyang Tao**[1,4,ζ,†]
[1]Duke University [2]Microsoft [3]Virginia Tech [4]Amazon [5]Fudan University [6]KAUST

Abstract

InfoNCE-based contrastive representation learners, such as SimCLR [1] and MoCo [2], have been tremendously successful in recent years. However, these contrastive schemes are notoriously resource demanding, as their effectiveness breaks down with small-batch training (*i.e.*, the $\log$-K curse, whereas K is the batch-size). In this work, we reveal mathematically why contrastive learners fail in the small-batch-size regime, and present a novel simple, non-trivial contrastive objective named FlatNCE, which fixes this issue. Unlike InfoNCE, our FlatNCE no longer explicitly appeals to a discriminative classification goal for contrastive learning. Theoretically, we show FlatNCE is the mathematical dual formulation of InfoNCE, thus bridging the classical literature on energy modeling; and empirically, we demonstrate that, with minimal modification of code, FlatNCE enables immediate performance boost independent of the subject-matter engineering efforts. The significance of this work is furthered by the powerful generalization of contrastive learning techniques, and the introduction of new tools to monitor and diagnose contrastive training. We substantiate our claims with empirical evidence on CIFAR10 and ImageNet datasets, where FlatNCE consistently outperforms InfoNCE.

1 Introduction

Due to their superior effectiveness [3, 4], easy implementation [5], and strong theoretical connection to mutual information (MI) estimation [6], *contrastive representation learning* has gained considerable momentum in recent years [7–11], especially in self-supervised learning setups [2, 1, 12, 13]. Despite encouraging progress, there are still many unresolved issues with contrastive learning, with the following three particularly relevant to this investigation:

(i) contrastive learners need a very large number of negative samples to work well [6];
(ii) the bias, variance, and performance tradeoffs are in debate [14, 15];
(iii) the lack of principled training diagnostic tools.

Among these three issues, (i) is most concerning: it implies training can be very expensive, and the needed massive computational resources may not be widely available. This has largely limited potential applications with more complex models or in budgeted scenarios. Even when such computational resources are accessible, the costs are prohibitive, and arguably entails a large carbon footprint. Consequently, large-scale contrastive training has essentially become a “rich man’s club”, with only resource affluent institutions in the game.

We believe addressing (ii) and (iii) holds the key to resolving (i), promising more affordable training of self-supervised learning for smaller teams and broader applications. Motivating our development is the major inconsistency between theory and practice is that, contrary to expectation, more biased estimators such as InfoNCE work better in practice than their tighter counterparts [16]. The prevailing conjecture is that these biased contrastive learners benefit from a lower estimation variance [14, 17]. However, this conjecture is mostly based on experimental observations rather than formal variance analyses [18], and the comparison is not technically fair since the less biased estimators use far

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

less samples [6, 19, 4]. Such incomplete understandings are partly caused by the absence of proper generic diagnostic tools to analyze contrastive learners. In this study, we hope to improve both the understanding and practice of contrastive representation learning via bridging these gaps.

Our development starts with a critical insight that challenges current beliefs: the poor learning efficiency of `InfoNCE`-based contrastive learners at small-batch regime is primarily due to floating-point overflow, not because of the larger bias it has suffered. This novel perspective motivates a simple, effective fix named `FlatNCE` that requires only one-line change of code relative to `InfoNCE`. Figure 1 visualizes our findings: while the `InfoNCE` and our `FlatNCE` are subjected to the same small-sample bias, the representation learned by the latter is much better. Apart from the above simple heuristic from an engineering perspective, we also back up our `FlatNCE` with solid mathematical analyses, unveiling its deep roots in statistical physics and convex optimization.

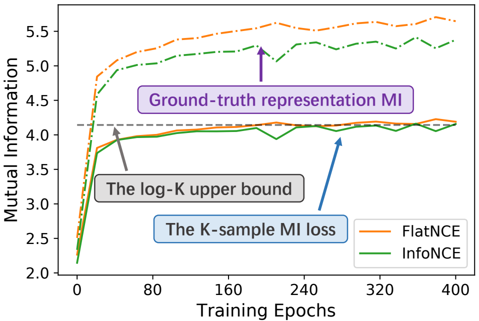

Figure 1: `FlatNCE` learns better representation when `InfoNCE` starts to struggle as mini-batch estimate approaches the log-K limit. Solid lines are the mini-batch estimate of representation MI, break lines are the ground-truth MI (higher is better), and the dashed line marks the log-K saturation.

Importantly, our research brings new insights into contrastive learning. We embrace an energy modeling view [20], and show appealing to a cross-entropy-based predictive objective as in `InfoNCE` is suboptimal. This echoes recent attempts in building non-discriminative contrastive learners [21], and to the best of our knowledge, we provide the first of its kind that comes with rigorous theoretical guarantees. Further, `FlatNCE` inspires a set of diagnostic tools that will benefit the contrastive learning community as a whole [14].

2 Why Is InfoNCE Failing at Small Batch-sizes (The log-K Curse)

Despite `InfoNCE`'s sweeping successes, here we provide a careful analysis to reveal that as the empirical `InfoNCE` estimate approaches saturation (*i.e.*, $\hat{I}_{\texttt{InfoNCE}} \to \log K$), its learning efficiency plunges due to limited numerical precision, which clarifies `InfoNCE`'s small-sample collapse.

Recall `InfoNCE` is a multi-sample mutual information estimator built on the idea of *noise contrastive estimation* (NCE) [22][1]. It was first described in [5] under the name *contrastive predictive coding* (CPC), and later formalized and coined `InfoNCE` in the work of [6]. Formally defined by

$$I^K_{\texttt{InfoNCE}}(X;Y|g) \triangleq \mathbb{E}_{(x_i,y_i)\sim p(x,y)}\left[\log \frac{\exp(g(x_i,y_i))}{\frac{1}{K}\sum_{j=1}^K \exp(g(x_i,y_j))}\right], \tag{1}$$

it constructs a formal lower bound to the mutual information, *i.e.*, $I^K_{\texttt{InfoNCE}}(X;Y|g) \leq I(X;Y)$. Here $g(x,y) \in \mathbb{R}$ is known as the *critic function* and K is the mini-batch size.

In practice the `InfoNCE` loss is computed from the `CrossEntropy` loss. For most deep learning platforms, the internal implementations exploit the `logsumexp` trick $\ell_{\text{CE}} = \texttt{logsumexp}(\{g_{ij}\}) - g_{ii} = \{\log(\sum_j \exp(g_{ij} - g_{\max})) + g_{\max}\} - g_{ii}$, to avoid numerical overflow, where $g_{ij} \triangleq g_\theta(x_i, y_j)$ and $g_{\max} \triangleq \max_j g_{ij}$. With a powerful learner for $g_\theta(x,y)$ and a small K such that $I(X;Y) > \log K$, we can reasonably expect $\hat{I}_{\texttt{InfoNCE}} \approx \log K$ after a few training epochs. Since g_{ii} itself is also included in the negative samples, this implies $g_{ii} \gg g_{ij}, \ \forall j \neq i$ almost always holds true, because $\hat{I}_{\texttt{InfoNCE}} = \log \frac{\exp(g_{ii})}{\frac{1}{K}\sum_j \exp(g_{ij})} \approx \log \frac{\exp(g_{ii})}{\frac{1}{K}\exp(g_{ii})} = \log K$, the contrast now becomes

$$\ell_{\text{CE}} = g_{ii} + \log(\sum\nolimits_j \exp(g_{ij} - g_{ii})) - g_{ii} = g_{ii} + \log(1 + o(1)) - g_{ii} \approx 0, \tag{2}$$

where $o(1)$ is the common notation for the terms that are small enough to be negligible.

This is where the `InfoNCE` algorithm becomes problematic: for low-precision floating-point arithmetics, *e.g.*, `float32` or `float16` as in standard deep learning applications, the relative numerical

[1] In some contexts, it is also known as *negative sampling* [23].

error is large when two similar numbers are subtracted from one another. The contrastive terms $g_{ij} - g_{ii}$, which are actually contributing the learning signals, will now be the $o(1)$ term that is engulfed by the dominating g_{ii}. In other words, `InfoNCE` has a low *signal-to-noise ratio* (SNR) when it approaches the $\log$-K saturation due to rounding errors (see Figure 2). With slight abuse of notation, we call it the $\log$-K curse[2].

3 Making It Flat: Fixing InfoNCE With Term-dropping FlatNCE

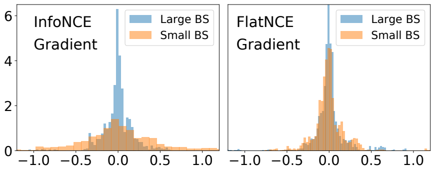

Figure 2: Comparison of model gradients with different batch-sizes (small=32, large=512). Consistent with our analyses, `InfoNCE` gradients show larger variance at small batch. Details see Appendix Section E.

Motivated by the discussions above, we build such a surrogate objective that overcomes `InfoNCE`'s difficulties. Incentivized by `InfoNCE`'s successes due to small estimating variance, we want the resulting objective to enjoy a low-variance profile. Taking this to the extreme, we propose **FlatNCE**, a *zero-variance* mini-batch, *self-normalized* contrastive objective implicitly optimizes MI

$$\mathrm{I}_{\texttt{FlatNCE}} = \frac{\sum_{j\neq i} \exp(g_\theta(x_i,y_j)-g_\theta(x_i,y_i))}{\texttt{detach}[\sum_{j\neq i} \exp(g_\theta(x_i,y_j)-g_\theta(x_i,y_i))]}, \quad (3)$$

where `detach[`$f_\theta(x)$`]` bars gradient back-propagation. For intuition, minimizing (3) heuristically forces a larger gap between the positive score $g(x_i, y_i)$ and negative score $g(x_i, y_j), i \neq j$, which is consistent with `InfoNCE`'s behavior. (3) explicitly excluded the "gold" positive pair (x_i, y_i), highlighting a key difference to `InfoNCE`, where the positive pair is intentionally retained. Note $I_{\texttt{FlatNCE}} \equiv 1$ for arbitrary inputs[3], albeit confusing this fulfills the zero-variance property. As an important remark, while the loss is zero-variance, the gradients are not.

We can in fact rigorously prove `FlatNCE` formally connects to `InfoNCE` and more broadly mutual information estimation (details relegated to our Appendix Section E). Mathematically, the gradient of `FlatNCE` equals to that of the `InfoNCE` also with the gold pair removed from the denominator (*i.e.*, the term-dropping `InfoNCE`). However, term-dropping `InfoNCE` essentially recovers the `TUBA` estimator which is known to associated with unstable training and poor performance in practice [6, 24]. This is more of numerical reasons that is overcome by the `FlatNCE` expression. What makes this particularly interesting is that `FlatNCE` can be considered the conjugate dual of `InfoNCE` in the view of convex optimization [25]. Specifically, via leveraging the Fenchel-Legendre duality trick [26, 27], we prove `FlatNCE` objective (3) recovers a batch-size independent tight MI bound recently proposed in [24], up to a non-gradient contributing term (which we put back when plotting Figure 1).

4 Effective Sample-size Scheduling For Contrastive Training

Existing contrastive training schemes use a temperature hyper-parameter β to tune the scale of $g_\theta = \beta \cdot \tilde{g}_\theta$, where $\tilde{g}$ is usually bounded between [-1,1] based on cosine similarity. β is known to be crucial for performance tuning, and currently there is no principled tuning heuristics. Our analyses above motivates us to introduce *Effective Sample-Size* (ESS), a normalized diagnostic statistics to monitor and tune contrastive training

$$\text{ESS} \triangleq 1/\left\{K \sum\nolimits_j w_j^2\right\} \in [1/K, 1], \text{ where } w_j = \frac{\exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))}{\sum_{j'\neq i} \exp(g_\theta(x_i, y_{j'}) - g_\theta(x_i, y_i))}. \quad (4)$$

A small ESS implies signal only comes from a small fraction of data (as in the `InfoNCE` failure case), and consequently less stable (see further analysis in Proposition S2). We hypothesized that contrastive learning would be most efficient if it draws signal from a more diverse sample pool, but does not absorb all info indiscriminately. That is to say, a moderate ESS should work best. These intuitions are verified in our experiments: Figure 6 shows for the best-performing temperature, `FlatNCE` has a larger ESS; and in Figure 7, we fixed ESS during training and confirm the model performs best for a moderate ESS$\in [0.2, 0.4]$. Please consult our Appendix Section E for details.

5 Self-supervised Learning Experiments

To validate our proposal, we benchmark `FlatNCE` against state-of-the-art solution `SimCLR` [1]. All experiments are implemented with `PyTorch` and executed on a maximal of at 4 NVIDIA V100 GPUs.

[2] In other contexts, the $\log$-K curse sometimes refers to the fact that the variance of a (sharp) non-parametric MI estimator grows exponentially wrt ground-truth MI [18].

[3] Note that the gradient of `FlatNCE` is not flat, that is why we can still optimize the representation.

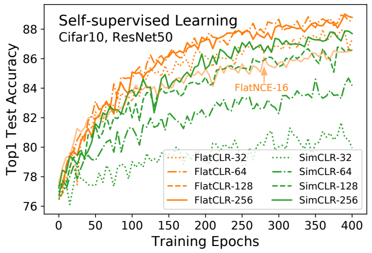

Figure 3: Sample efficiency comparison for `SimCLR` and `FlatCLR` on `Cifar10`.

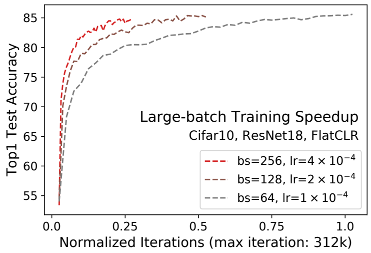

Figure 4: Speed up of large-batch training. Larger batch leads to faster convergence.

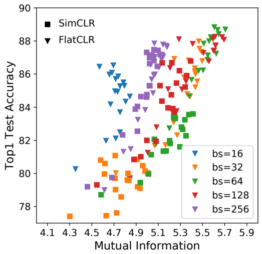

Figure 5: Representation MI strongly correlates with performance.

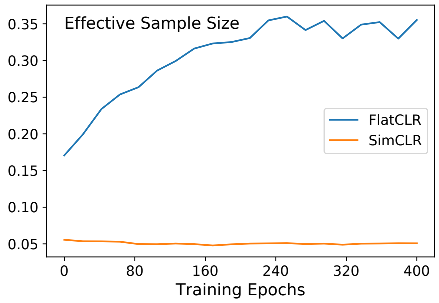

Figure 6: Evolution of ESS for Figure 1, `FlatNCE` receives learning signal from a more diverse pool of negative samples.

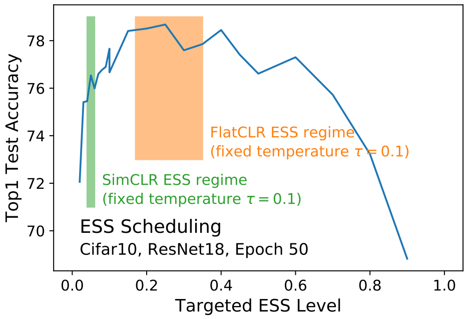

Figure 7: ESS scheduling results. Contrastive models learn most efficiently & achieves best performance with a moderate ESS (0.2-0.4).

Table 1: ImageNet SSL results.

	Epoch	10	20	30	40	50	60	70	80	90	100
Top-1 Acc	`SimCLR`	40.93	46.22	48.64	50.14	52.14	53.62	55.20	56.36	56.99	57.13
	`FlatCLR`	**42.40**	**47.69**	**49.96**	**52.27**	**54.11**	**55.48**	**56.98**	**58.21**	**58.80**	**59.74**
Top-5 Acc	`SimCLR`	65.34	70.92	73.63	75.38	76.90	78.24	79.59	80.58	80.85	81.00
	`FlatCLR`	**67.17**	**72.61**	**74.59**	**76.77**	**78.29**	**79.67**	**81.06**	**82.19**	**82.71**	**83.18**

Table 2: ImageNet SSL transfer learning results.

	Dataset	Cifar10	Cifar100	VOC2007	Flower	SUN397
Linear evaluation	`SimCLR`	87.74	65.40	69.38	90.03	49.62
	`FlatCLR`	87.92	65.76	69.66	90.23	51.31
Fine-tune	`SimCLR`	94.61	76.67	69.57	93.58	56.97
	`FlatCLR`	**95.50**	**78.92**	**70.73**	**95.02**	**58.37**

Setup. Our follow the settings in [1] and our codebase is modified from a public `PyTorch` `SimCLR` implementation[4]. Specifically, we train 256-D feature by maximizing the self-MI between two random views of data, and report the test set classification accuracy using a linear classifier trained to convergence. We report performance based on `ResNet-50`, and analyze the learning dynamics using the smaller `ResNet-18` for quick iterations. For detailed settings consult our Appendix Section I.

Main results. In Figure 3, we evaluate the quality of representations learned with different batch-sizes using `ResNet-50` on `Cifar10`. `FlatCLR` showed superior small-batch efficiency: with only BS=32, `FlatCLR` matches the performance of `SimCLR` at BS=256, which is an 8× boost. Figure 4 showed we can use linear learning rate scaling to speedup convergence with larger batch-sizes. Figure 5 showed ground-truth representation self-MI strongly correlates with model performance, which proved tighter bounds give worse performance only when they are not effectively optimizing the representation. Finally, we show `ImageNet` SSL results in Table 1, and SSL transfer learning results in Table 2, where `FlatCLR` leads consistently. More results and analyses are in our Appendix Section I.

[4] `https://github.com/sthalles/SimCLR`

References

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *ICML*, 2020.

[2] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *CVPR*, 2020.

[3] N. Tishby and N. Zaslavsky, “Deep learning and the information bottleneck principle,” in *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2015.

[4] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, “Learning deep representations by mutual information estimation and maximization,” in *ICLR*, 2019.

[5] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.

[6] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, “On variational bounds of mutual information,” in *ICML*, PMLR, 2019.

[7] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in *CVPR*, 2018.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, “Learning transferable visual models from natural language supervision,” *arXiv preprint arXiv:2103.00020*, 2021.

[9] M. Laskin, A. Srinivas, and P. Abbeel, “Curl: Contrastive unsupervised representations for reinforcement learning,” in *International Conference on Machine Learning*, pp. 5639–5650, PMLR, 2020.

[10] U. Gupta, A. Ferber, B. Dilkina, and G. V. Steeg, “Controllable guarantees for fair outcomes via contrastive information estimation,” *arXiv preprint arXiv:2101.04108*, 2021.

[11] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig, “Scaling up visual and vision-language representation learning with noisy text supervision,” *arXiv preprint arXiv:2102.05918*, 2021.

[12] I. Misra and L. v. d. Maaten, “Self-supervised learning of pretext-invariant representations,” in *CVPR*, 2020.

[13] T. Gao, X. Yao, and D. Chen, “Simcse: Simple contrastive learning of sentence embeddings,” *arXiv preprint arXiv:2104.08821*, 2021.

[14] S. Arora, H. Khandeparkar, M. Khodak, O. Plevrakis, and N. Saunshi, “A theoretical analysis of contrastive unsupervised representation learning,” *ICML*, 2019.

[15] K. Nozawa and I. Sato, “Understanding negative samples in instance discriminative self-supervised representation learning,” *arXiv preprint arXiv:2102.06866*, 2021.

[16] M. Tschannen, J. Djolonga, P. K. Rubenstein, S. Gelly, and M. Lucic, “On mutual information maximization for representation learning,” *ICLR*, 2020.

[17] J. Song and S. Ermon, “Understanding the limitations of variational mutual information estimators,” in *ICLR*, 2020.

[18] D. McAllester and K. Stratos, “Formal limitations on the measurement of mutual information,” *arXiv preprint arXiv:1811.04251*, 2018.

[19] P. Bachman, R. D. Hjelm, and W. Buchwalter, “Learning representations by maximizing mutual information across views,” in *NeurIPS*, 2019.

[20] M. Arbel, L. Zhou, and A. Gretton, “Generalized energy based models,” in *ICLR*, 2021.

[21] T. Wang and P. Isola, “Understanding contrastive representation learning through alignment and uniformity on the hypersphere,” in *ICML*, PMLR, 2020.

[22] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *AISTATS*, pp. 297–304, 2010.

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

[24] Q. Guo, J. Chen, D. Wang, Y. Yang, X. Deng, F. Li, L. Carin, and C. Tao, “Tight mutual information estimation with contrastive Fenchel-Legendre optimization,” 2021. [Available online; accessed 28-May-2021].

[25] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[26] W. Fenchel, “On conjugate convex functions,” *Canadian Journal of Mathematics*, vol. 1, no. 1, pp. 73–77, 1949.

[27] C. Tao, L. Chen, S. Dai, J. Chen, K. Bai, D. Wang, J. Feng, W. Lu, G. Bobashev, and L. Carin, “On Fenchel mini-max learning,” in *NeurIPS*, 2019.

[28] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[29] M. U. Gutmann and A. Hyvärinen, “Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics,” *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 307–361, 2012.

[30] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS*, 2013.

[31] A. Mnih and K. Kavukcuoglu, “Learning word embeddings efficiently with noise-contrastive estimation,” in *NIPS*, vol. 26, pp. 2265–2273, 2013.

[32] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *SIGKDD*, 2014.

[33] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in *SIGKDD*, 2016.

[34] P. H. Le-Khac, G. Healy, and A. F. Smeaton, “Contrastive representation learning: A framework and review,” *IEEE Access*, 2020.

[35] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, “Contrastive learning with hard negative samples,” *arXiv preprint arXiv:2010.04592*, 2020.

[36] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, “Hard negative mixing for contrastive learning,” *arXiv preprint arXiv:2010.01028*, 2020.

[37] C.-Y. Chuang, J. Robinson, L. Yen-Chen, A. Torralba, and S. Jegelka, “Debiased contrastive learning,” in *NeurIPS*, 2020.

[38] L. Logeswaran and H. Lee, “An efficient framework for learning sentence representations,” *arXiv preprint arXiv:1803.02893*, 2018.

[39] S. Ozair, C. Lynch, Y. Bengio, A. v. d. Oord, S. Levine, and P. Sermanet, “Wasserstein dependency measure for representation learning,” *NeurIPS*, 2019.

[40] O. Henaff, “Data-efficient image recognition with contrastive predictive coding,” in *ICML*, PMLR, 2020.

[41] M. Wu, C. Zhuang, D. Yamins, and N. Goodman, “On the importance of views in unsupervised representation learning,” *preprint*, vol. 3, 2020.

[42] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*, pp. 177–186, Springer, 2010.

[43] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” *arXiv preprint arXiv:1609.04836*, 2016.

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

[45] Y. Burda, R. Grosse, and R. Salakhutdinov, “Importance weighted autoencoders,” in *ICLR*, 2016.

[46] T. Rainforth, T. A. Le, M. I. C. J. Maddison, and Y. W. T. F. Wood, “Tighter variational bounds are not necessarily better,” in *NIPS workshop*, 2017.

[47] A. Hyvarinen, H. Sasaki, and R. Turner, “Nonlinear ica using auxiliary variables and generalized contrastive learning,” in *AISTATS*, pp. 859–868, 2019.

[48] M. D. Donsker and S. S. Varadhan, “Asymptotic evaluation of certain markov process expectations for large time. iv,” *Communications on Pure and Applied Mathematics*, vol. 36, no. 2, pp. 183–212, 1983.

[49] X. Nguyen, M. J. Wainwright, and M. I. Jordan, “Estimating divergence functionals and the likelihood ratio by convex risk minimization,” *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5847–5861, 2010.

[50] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, “Mutual information neural estimation,” in *ICML*, PMLR, 2018.

[51] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of convex analysis*. Springer Science & Business Media, 2012.

[52] H. Robbins and S. Monro, “A stochastic approximation method,” *The annals of mathematical statistics*, pp. 400–407, 1951.

[53] D. Barber and F. Agakov, “The IM algorithm: a variational approach to information maximization,” *Advances in neural information processing systems*, vol. 16, p. 201, 2004.

Appendix

Table of Contents

A The Staggering Cost of Training Contrastive Learners

In Table S1 we summarize the associated cost of training state-of-the-art contrastive learners. We have used the numbers from the original papers to compute the cost. The number of devices and time of training for the largest model reported in the respective papers are used, while we use the online quotes from Google Cloud (for TPU units) and Amazon AWS (for GPU units) for the hourly cost of dedicated computing devices. We only focused on the computation cost, so the potential charges from storage and network traffic are omitted. Note that this table only reports the number of computing devices used in the final training where all parameters have been tuned to optimal, the actual expenditures associated with the development of these models can be significantly higher. Usually researchers and engineers spent more time tuning the parameters and exploring ideas before finally come up with a model that can be publicized. Also, the cost for performance evaluation is not count towards the cost, and some of the papers have employed intensive grid-search of parameters for evaluation, which in our experience can be even more costly than training the contrastive learners at times. And we do find fine-tuning evaluation can drastically boost the performance metrics.

Table S1: Cost of training a contrastive learner

Model	Sponsor	Neg. Size	Infrastructure	Train Time	Est. Cost
`MoCo` [2]	Facebook	$65,536$	64 V100 GPUs	6 days	$23k
`SimCLR` [1]	Google	$4,096$	128-core TPU-v3	15 hours	$1,720
`CLIP` [8]	OpenAI	$32,768$	592 V100 GPUs	18 days	$634k

B More Background On Contrastive Learning

As a consequence of the superior effectiveness in self-supervised learning setups [3, 4] and their relatively easy implementation [5], *contrastive representation learning* has gained considerable momentum in recent years. Successful applications have been reported in computer vision [7, 2, 1, 12], natural language processing [13, 8], reinforcement learning [9], fairness [10], amongst many others.

Originally developed for nonparametric density estimation, the idea of learning by contrasting *positive* and *negative* samples has deep roots in statistical modeling [28]. In the seminal work of [29], its connection to discriminative classification was first revealed, and early utilization of the idea was celebrated by the notable success in training word embeddings [30, 31]. Framed under the name *negative sampling* [23], contrastive techniques have been established as indispensable tools in scaling up the learning of intractable statistical models such as graphs [32, 33].

More recently, surging interest in contrastive learners was sparked by the renewed understanding that connects to *mutual information* estimation [5, 6]. Fueled by the discovery of efficient algorithms and strong performance [1], extensive research has been devoted to this active topic [34]. These efforts range from theoretical investigations such as generalization error analyses [14] and asymptotic characterizations [21], to more practical aspects including hard-negative reinforcement [35, 36], and sampling bias adjustment [37]. Along with various subject matter improvements [38–41, 13, 8], contrastive learners now provide comprehensive solutions for self-supervised learning.

C Rethinking Contrastive Learning: Experimental Evidence & Discussions

We contribute this section to the active discussions on some of the most important topics in contrastive learning.

Our discussions will be grounded on the new experimental results from `Cifar10` with a `ResNet` backbone, with a `PyTorch` codebase of the `InfoNCE`-backed `SimCLR` and its `FlatNCE` counterpart `FlatCLR`. Note instead of trying to set new performance records (because of limited computational resources in our university setting), experiments in this section are designed to reveal important aspects of contrastive learning, and to ensure our results can be easily reproduced with reasonable computation resources.

Breaking the curse, small-batch contrastive learning revived. We show that with our novel `FlatNCE` objective, successful contrastive learning applications are no longer exclusive to the costly large-batch training. In Figure 3 we see pronounced small-sample performance degradation for `SimCLR`, while the `FlatCLR` is far less sensitive to the choice of batch size. In fact, we see `FlatCLR`-16 matches performance of its `SimCLR`-128 counterpart, corresponding to an $8\times$ boost in efficiency. And in all cases `FlatCLR` consistently works better compared to the same-batch-size `SimCLR`. Despite the encouraging improvements in the small-batch regime, large-batch training does provide better results for both `SimCLR` and `FlatCLR`. Additionally, leveling up parallelism greatly reduces the overall training time (Figure 4), as a larger batch-size enables stable training with a larger learning rate [42–44] [5]. The main merits of our result are: (i) the enabling of contrastive learning for very budgeted applications, where large-batch learning is prohibitive; and (ii) consistent improvement over `InfoNCE`, especially wrt the cost-performance trade-off.

Is tighter MI bound actually better or worse? An interesting observation made by a few independent studies is that, perhaps contrary to expectation, tighter bounds on MI do not necessarily lead to better performance on the downstream tasks [16]. To explain this, existing hypotheses have focused on the variance and sample complexity perspectives [17]. To address this, we compare the actual MI [6] to the mini-batch MI estimate, and plot the respective typical training curves in Figure 1. Since `FlatNCE` itself is not associated with a number to bound MI (because it is theoretically tight), we use an `InfoNCE` estimate based-on the `FlatNCE` representation. Observe that although the sample MI estimates are tied, `FlatCLR` robustly improves the ground-truth MI as `SimCLR` approaches the $\log$-K saturation point and become stagnant. To further understand how MI relates to downstream performance, we plot the Top-1 accuracy against the true MI using all our model training checkpoints

[5] While learning rate scheduling does affect performance, it is beyond the scope of our current investigation.

[6] Ground-truth MI is approximated by `InfoNCE` using a very large negative sample pool (100X mini-batch).

(Figure 5), and confirm a strong linear relation between the two (Pearson correlation $\rho = 0.65$, p-value $< 10^{-20}$). However, this link is not evident using the mini-batch sample MI (Figure S1).

ESS for monitoring and tuning contrastive learning. As an important tool introduced in this work, we want to demonstrate the usefulness of ESS in contrastive training. Figure 6 plots ESS curves for the training dynamics described in Figure 1, and we see drastically different profiles. As predicted by our analyses, `SimCLR`'s ESS monotonically decreases as it approaches the `InfoNCE` saturation (from 0.06 to 0.05), while `FlatNCE`-ESS instead climbs up ($0.17 \to 0.35$). The performance gap widens as the ESS difference becomes larger, thus confirming the superior sample efficiency of `FlatNCE`. Next we experimented with ESS-scheduling: instead of a fixed temperature, we fix the ESS throughout training, and then compare model performance. Figure 7 shows a snapshot of training progress per targeted ESS value at epoch 50, where the estimated MI just started to plateau. The result indicates ESS range $[0.15, 0.4]$ works well for `Cifar10`, while `SimCLR` with fixed temperature only covers the sub-optimal $[0.04, 0.06]$. These interesting observation warrant further future investigations on ESS control in contrastive training.

Self-normalized contrastive learning as constrained optimization. Here we want to promote a new view, which considers self-normalized contrastive learning as a form of constrained optimization. In this view, including multiple negative samples in the update of the critic function is necessary for contrastive learning. This conclusion comes from our numerous failed attempts in designing alternative few-sample contrastive learning objectives that simultaneously reduce estimation variance and tighten the MI bound. Since the feature encoders are usually built with complex neural networks, the representations can be rather sensitive to the changes in encoder parameters. So while the gradient update direction may maximally benefit the MI estimate, it may disrupt the representation and thus compromise the validity of the variational MI estimate. Including negative samples in the updates of the critic g_θ allows the use of negative samples to provide instant feedback on which directions are bad, and to steer away from. More negative samples (*i.e.*, a larger K) will enforce a more confined search space, thus allowing the critic updates to proceed more confidently with larger learning rates. Also, comparison should be made to *importance-weighted variational auto-encoder* (IW-VAE) [45], which also leverages a self-normalized objective for representation learning and inference. However, IW-VAE has been proven harmful to representation learning, although it provably tightens the likelihood bound [46]. Finally, our new approach also promises to scale up & improve *generalized contrastive learning* [47].

Connections to variational mutual information estimation. Table S2 summarizes representative examples of nonparametric variational MI bounds in the literature, whose difference can be understood based on how information from negative samples are aggregated. Before `InfoNCE`, *Donsker-Varadhan* (`DV`) [48] and *Nguyen-Wainwright-Jordan* (`NWJ`) [49] are the most widely practiced MI estimators. `NWJ` is generally considered non-contrastive as positive and negative samples are compared, respectively, at $\log$ and $\exp$ scales. `DV` differs from `InfoNCE` by excluding the positive sample from the negative pool, which is similar to the practice of our `FlatNCE`. However, `DV` is numerically unstable and necessitates careful treatment to be useful [50]. Also note some literature had unfairly compared the the multi-sample `InfoNCE` to the single-sample versions of its competitors, partly because the alternatives do not have efficient multi-sample implementations. To the best of our knowledge, closest to this research is the concurrent work of [24], where the contrastive *Fenchel-Lengendre* estimator is derived. While developed independently from completely different perspectives, `FlatNCE` enjoys the duality view promoted by [24] and inherits all its appealing theoretical properties. Our theoretical and empirical results complemented nicely the theories from [24].

D Contrastive Representation Learning with InfoNCE

Proposition S1. `InfoNCE` is an asymptotically tight lower bound to the mutual information, *i.e.*,

$$I(X;Y) \geq I^K_{\texttt{InfoNCE}}(X;Y|f), \quad \lim_{K\to\infty} I^K_{\texttt{InfoNCE}}(X;Y) \to I(X;Y). \tag{5}$$

Proof. See [6] for a neat proof on how the multi-sample `NWJ` upper bounds `InfoNCE`. Since `NWJ` is a lower bound to MI, `InfoNCE` also lower bounds MI.

What remains is to show the `InfoNCE` bound is asymptotically tight. We only need to prove that with a specific choice of $f(x,y)$, `InfoNCE` recovers $I(X;Y)$. To this end, let us set $f(x,y) = f^*(x,y) =$

Table S2: Comparison of representative variational MI objectives. We use (x_i, y_i) to denote the positive sample drawn from the joint density $p(x, y)$, (x_i, y_j) where $j \neq i$, for the negative samples from $p(x)p(y)$, and $m(x, y^{1:K}) \triangleq \frac{1}{K} \sum_{j=1}^{K} \exp(g(x_i, y_j))$. See the following table for more details.

Name	Objective	Bias	Stability
Donsker-Varadhan [48]	$g(x_{,i}\, y_i) - \log(\sum_{j=1}^{K} \exp(g(x_i, y_j))/K)$	Large	Poor
Nguyen-Wainwright-Jordan [49]	$g(x_i, y_i) - \sum_{j=1}^{K} \exp(g(x_i, y_j)/K$	Low	Okay
Fenchel-Legendre [24]	$u(x_i, y_i) + \sum_{j=1}^{K} \exp(-u(x_i, y_j) + g(x_i, y_j) - g(x_i, y_i))/K$	Low	Okay
`InfoNCE` [5]	$g(x_i, y_i) - \log(m(x_i, \{y_i, y_j^{1:K-1}\}))$	Large	Excellent
`FlatNCE` (Ours)	$\{m(x, y_j^{1:K}) - g(x_i, y_i)\}/\texttt{detach}[\{m(x_i, y_j^{1:K}) - g(x_i, y_i\}]$	Low	Excellent

$\frac{p(y|x)}{p(y)}$, and we have

$$\begin{align*}
I_{\texttt{InfoNCE}}^{K}(f^*) &= \mathbb{E}_{p^K}\left[\log\left(\frac{f^*(x_k, y_k)}{f^*(x_k, y_k) + \sum_{k' \neq k} f^*(x_k, y_{k'})}\right)\right] + \log K \tag{6}\\
&= -\mathbb{E}\left[\log\left(1 + \frac{p(y)}{p(y|x)} \sum_{k'} \frac{p(y_{k'}|x_k)}{p(y_{k'})}\right)\right] + \log K \tag{7}\\
&\approx -\mathbb{E}\left[\log\left(1 + \frac{p(y)}{p(y|x)}(K-1)\mathbb{E}_{y_{k'}} \frac{p(y_{k'}|x_k)}{p(y_{k'})}\right)\right] + \log K \tag{8}\\
&= -\mathbb{E}\left[\log\left(1 + \frac{p(y_k)}{p(y_k|x_k)}(K-1)\right)\right] + \log K \tag{9}\\
&\approx \underbrace{-\mathbb{E}\left[\log \frac{p(y)}{p(y|x)}\right]}_{I(X;Y)} - \log(K-1) + \log K \tag{10}\\
& \tag{11}
\end{align*}$$

Now taking $K \to \infty$, the last two terms cancels out. □

A few technical remarks are useful for our subsequent developments: (i) the K-sample `InfoNCE` estimator is upper bounded by $\log K$; (ii) in practice, `InfoNCE` is implemented with the `CrossEntropy` loss for multi-class classification, where $f(x, y)$ is parameterized by its logit $g_\theta(x, y) = \log f(x, y)$; ($iii$) optimizing for $f(x, y)$ tightens the bound, and the bound is sharp if $f(x, y) = p(x|y)e^{c(x)}$, where $c(x)$ is an arbitrary function on $\mathcal{X}$; and (iv) `InfoNCE`'s successes have been largely credited to the fact that its empirical estimator has much smaller variance relative to competing solutions.

E FlatNCE and Generalized Contrastive Representation Learning

E.1 Understanding FlatNCE

We first define the following variant of the `FlatNCE`

$$I_{\texttt{FlatNCE}}(g_\theta) = \frac{1 + \sum_j \exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))}{1 + \texttt{detach}[\sum_j \exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))]}. \tag{12}$$

Note that $I_{\texttt{FlatNCE}}(g_\theta)$ corresponds to adding the positive sample y_i to the set of negative samples, because the zero contrast of the positive sample always gives the constant one. The following statement verifies $I_{\texttt{FlatNCE}}(g_\theta)$ is equivalent to `InfoNCE` in terms of differentiable optimization.

Proposition S1. $\nabla_\theta I_{\texttt{FlatNCE}}(g_\theta) = \nabla_\theta I_{\texttt{InfoNCE}}(g_\theta)$.

Proof. Without loss of generality we denote y_0 as the positive sample and all $y_j, j > 0$ as the negative samples. Recall

$$\begin{aligned} & \texttt{CrossEntropyLoss}(\texttt{logits} = g_\theta(x_0, y_j), \texttt{label} = 0) && (13) \\ = & -\log \frac{\exp(g_\theta(x_0, y_0))}{\sum_j \exp(g_\theta(x, y_j))} && (14) \\ = & \log \sum\nolimits_j \exp(g_\theta(x_0, y_j) - g_\theta(x_0, y_0)) && (15) \end{aligned}$$

Since $\nabla \log f = \frac{\nabla f}{f}$, so

$$\nabla_\theta I_{\texttt{FlatNCE}}(g_\theta) = \nabla_\theta \mathcal{L}_{\texttt{CrossEntropy}} = \frac{\nabla_\theta \{\sum_j \exp(g_\theta(x_0, y_j) - g_\theta(x_0, y_0))\}}{\sum_j \exp(g_\theta(x_0, y_j) - g_\theta(x_0, y_0))} = \nabla_\theta I_{\texttt{InfoNCE}}(g_\theta) \tag{16}$$

which concludes our proof (we omit the sign here for brevity). □

Equation (12) and Proposition S1 indicate how InfoNCE and FlatNCE are connected. Based on this, we continue our discussion on why InfoNCE fails for small batch sizes. We start by showing the gradient of FlatNCE and its variant $I_{\texttt{FlatNCE}}$ (so equivalently, InfoNCE) is given by a self-normalized importance-weighted gradient estimator, as formalized below.

Proposition S2. The gradient of FlatNCE is an importance-weighted estimator of the form

$$\nabla I_{\texttt{FlatNCE}} = \sum\nolimits_j w_j \nabla g_\theta(x_i, y_j) - \nabla g_\theta(x_i, y_i), \quad \text{where } w_j = \frac{\exp(g_\theta(x_i, y_j))}{\sum_{j'} \exp(g_\theta(x_i, y'_{j'})))}. \tag{17}$$

Proof. Let us pick up from (16) from last proof, we have

$$\begin{aligned} \nabla I_{\texttt{FlatNCE}} &= \frac{\nabla_\theta \{\sum_j \exp(g_\theta(x_i, y_j) - g_\theta(x_0, y_i))\}}{\sum_j \exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))} && (18) \\ &= \frac{\sum_j \exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))(\nabla_\theta \{g_\theta(x_0, y_j) - g_\theta(x_i, y_i)\})}{\sum_j \exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))} && (19) \\ &= \sum\nolimits_j w_j \nabla_\theta g_\theta(x_i, y_j) - (\sum\nolimits_j w_j) \nabla_\theta g_\theta(x_i, y_i) && (20) \\ &= \sum\nolimits_j w_j \nabla_\theta g_\theta(x_i, y_j) - g_\theta(x_i, y_i) && (21) \end{aligned}$$

here $w_j \triangleq \exp(g_\theta(x_i, y_j))/(\sum_{j'} \exp(g_\theta(x_i, y_{j'})))$, as the term $\exp(-g_\theta(x_i, y_i))$ has been canceled out. □

When $\hat{I}_{\texttt{InfoNCE}}$ approaches $\log K$, we know $w_i \approx 1, w_{j \neq i} \approx 0$, and consequently $\nabla I_{\texttt{InfoNCE}} \approx \nabla g_\theta(x_i, y_i) - \nabla g_\theta(x_i, y_i) = 0$.

Consequently, as long as the positive sample is in the denominator the learning signal vanishes. What makes matters worse, the low-precision computations employed to speed up training introduce rounding errors, further corrupting the already weak gradient. On the other hand, in FlatNCE larger weights will be assigned to the more challenging negative samples in the batch, thus prioritizing hard negatives.

Proposition S2 also sheds insights on temperature annealing. Setting $\beta \neq 1$ re-normalizes the weights by exponential scaling (*i.e.*, $w_j(\beta) = w_j^\beta / \sum_{j'} w_{j'}^\beta$). So the optimizer will focus more on the hard negative samples at a lower temperature (*i.e.*, larger β), while for a higher temperature it treats all negative samples more equally. This new gradient interpretation reveals that β affects the learning dynamics in addition to the well-known fact that β modulates MI bound tightness.

Lastly, to fill in an important missing piece, we prove that FlatNCE is a formal MI lower bound.

Lemma S3. For $\{(x_j, y_j)\}_{j=1}^K$, let $I^K_{\texttt{InfoNCE}}(g_\theta) \triangleq -\log \frac{1}{K} \sum_j \exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))$. Then for arbitrary $u \in \mathbb{R}$, we have inequality

$$I^K_{\texttt{InfoNCE}}(g_\theta) \geq 1 - u - \frac{1}{K} \sum\nolimits_j \exp(-u + g_\theta(x_i, y_j) - g_\theta(x_i, y_i)), \tag{22}$$

and the equality holds when $u = \frac{1}{K} \sum_j \exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))$.

Our proof is inspired by the technique used in [27] for non-parametric likelihood approximations, which is based on the celebrated Fenchel-Legendre duality given below.

Definition S4 (Fenchel-Legendre duality [26])**.** Let $f(t)$ be a proper convex, lower-semicontinuous function; then its convex conjugate function $f^*(v)$ is defined as $f^*(v) = \sup_{t\in\mathcal{D}(f)}\{tv - f(t)\}$, where $\mathcal{D}(f)$ denotes the domain of function f [51]. We call $f^*(v)$ the *Fenchel-Legendre conjugate* of $f(t)$, which is again convex and lower-semicontinuous. The Fenchel-Legendre conjugate pair (f, f^*) are dual to each other, in the sense that $f^{**} = f$, *i.e.*, $f(t) = \sup_{v\in\mathcal{D}(f^*)}\{vt - f^*(v)\}$.

Example. The Fenchel-Legendre dual for $f(t) = -\log(t)$ is $f^*(v) = -1 - \log(-v)$.

Proof. Let us write `InfoNCE` as

$$I_{\texttt{InfoNCE}}(g) = -\log \sum\nolimits_j \exp(g_\theta(x_0, y_j) - g_\theta(x_0, y_0)). \tag{23}$$

Replacing the $-\log(t)$ term in $I_{\texttt{InfoNCE}}(t)$ with its Fenchel-Legendre dual $-1 - \log(-v)$, then Proposition is immediate after properly rearranging the terms and write $u = -\log v$. □

What makes this particularly interesting is that `FlatNCE` can be considered the conjugate dual of `InfoNCE`. In convex analysis, u and g in (22) are known as the *Fenchel conjugate pair* [26, 27, 24]. By taking the expectation wrt $p^K(x, y)$ and setting $u(\{(x_j, y_j)\})$ to its optimal value, we essentially recover $I_{\texttt{FlatNCE}}(g_\theta)$: the only difference to the conjugate of `InfoNCE` is the term $(1 - u)$ which is considered fixed and does not participate in optimization. As such, the following Corollary is immediate[7].

Corollary S5. $I^K_{\texttt{FlatNCE}}(g_\theta) = I^K_{\texttt{InfoNCE}}(g_\theta) \leq I(X;Y),\ \ I^K_{\texttt{FlatNCE}}(g_\theta) \leq I(X;Y).$

To make our proof simpler, we follow some theoretical results developed in [24], included below for completeness.

Proposition S6 (The Fenchel-Legendre Optimization Bound, Proposition 2.2 in [24])**.**

$$I_{\texttt{FLO}}(u, g) \triangleq \left\{\mathbb{E}_{p(x,y)p(y')}\left[u(X, Y) + \exp(-u(X, Y) + g(X, Y') - g(X, Y))\right]\right\} + 1 \tag{24}$$

$$I(X;Y) = -\min\nolimits_{u,g}\{I_{\texttt{FLO}}(u, g)\} \tag{25}$$

Sketch of proof for Proposition S6. Recall the *Donsker-Varadhan* (DV) bound [48] is given by

$$I_{\texttt{DV}} \triangleq \mathbb{E}_{p(x,y)}[g(x, y) - \log(\mathbb{E}_{p(y')}[\exp(g(x, y'))])]. \tag{26}$$

Then we proceed similarly to the proof of Lemma S3.

Remark. Here we consider $g(x, y)$ as the primal critic and $u(x, y)$ as the dual critic. Since arbitrary choice of primal/dual critics always lower bounds MI, we can either jointly optimize the two critics, or train in an iterative fashion: optimize one at a time while keep the other fixed. Let us consider the case u is fixed and only update g, the proof below shows with an appropriate choice of u, Corollary 3.4 follows.

Proof of Corollary S5
Given $g_\theta(x, y)$ and empirical samples $\{(x_j, y_j)\}$, let us set $u(x, y)$ to

$$\hat{u}^*(g_\theta) = \log\left(\frac{1}{K}\sum\nolimits_j \exp(g_\theta(x_i, y_j) - g_\theta(x_i, y_i))\right) \tag{27}$$

Plug $(g_\theta, \hat{u}^*)$ into the right hand side of Equation (9) proves $\hat{u}^* + I_{\texttt{FlatNCE}} - 1$ lower bounds mutual information. Since $\hat{u}^*$ does not contribute gradient, we can consider $I_{\texttt{FlatNCE}} \leq I(X;Y)$ holds up to a constant term. In other words, we are effectively optimizing a lower bound to MI, although $I_{\texttt{FlatNCE}}$ does not technically a lower bound – this is still OK since the difference does not contribute learning signal. □

[7] Using a similar technique, we can also show (3) lower bounds mutual information.

E.2 Generalizing FlatNCE

The formulation of `FlatNCE` enables new possibilities for extending contrastive representation learning beyond its original form. In this section, we discuss some generalizations that make contrastive learning more flexible, including new tools for training diagnosis and tuning.

Hölder FlatNCE. To further generalize contrastive learning, we re-examine the objective of `FlatNCE`. A key observation is that the numerator aggregates individual *evidence* of MI from the negative samples $(x, y') \sim p(x)p(y')$ through the critic function $g_\theta(x, y)$, with arithmetic mean. Possibilities are that if we change the aggregation step, we also change how it learns MI in a way similar to the importance weighting perspective discussed above. This inspires us to consider the more general aggregation procedures, such as the Hölder mean defined below.

Definition S7 (Hölder mean)**.** For $\{a_i \in \mathbb{R}_+\}_{i=1^n}$ and $\gamma \in \mathbb{R}$, the Hölder mean is defined as $m_\gamma(\{a_i\}_{i=1}^n) = \left(\frac{1}{n}\sum_i a_i^\gamma\right)^{\frac{1}{\gamma}}$.

Note Hölder mean recovers many common information pooling operations, such as $\min$ ($\gamma = -\infty$), $\max$ ($\gamma = \infty$), geometric mean ($\gamma \to 0$), root mean square ($\gamma = 2$), and arithmetic mean ($\gamma = 1$) as employed in our `FlatNCE`. This allows us to define a new family of contrastive learning objectives.

Definition S8 (Hölder-`FlatNCE`)**.**

$$I_\gamma \triangleq \sum_i \frac{m_\gamma(\{\exp(g_{ij} - g_{ii}\}_j)}{\texttt{detach}[m_\gamma(\{\exp(g_{ij} - g_{ii}\}_j)]}. \tag{28}$$

The following Proposition shows that Hölder-`FlatNCE` is equivalent to annealed `FlatNCE`.

Proposition S9. $\nabla I_\gamma(g_\theta) = \frac{1}{\gamma}\nabla I_{\texttt{FlatNCE}}(\gamma \cdot g_\theta)$.

Proof. Denoting $f_j = \exp(g_j)$, and we have

$$\nabla I_\gamma(g_\theta) = \frac{\nabla m_\gamma(\{f_j\})}{m_\gamma(\{f_j\})} \tag{29}$$

$$= \frac{\frac{1}{\gamma}(\frac{1}{n}\sum_j f_j^\gamma)^{\frac{1}{\gamma}-1}\{\gamma\frac{1}{n}\sum_j f_j^{\gamma-1}\nabla f_j\}}{(\frac{1}{n}\sum_j f_j^\gamma)^{\frac{1}{\gamma}}} \tag{30}$$

$$= \frac{\sum_j f_j^{\gamma-1}\nabla f_j}{\sum_j f_j^\gamma} \tag{31}$$

$$= \frac{\nabla \sum_j \exp(\gamma g_j)}{\gamma \sum_j \exp(\gamma g_j)} \tag{32}$$

$$= \frac{1}{\gamma}\nabla I_{\texttt{FlatNCE}}(\gamma \cdot g_\theta) \tag{33}$$

□

As an important remark, we note the sample gradient of `FlatNCE` is a (randomly) re-scaled copy of the true gradient (normalized by Z_θ instead of $\hat{Z}_\theta$), so we are still optimizing the model in the right direction using stochastic gradient descent (SGD) [52]. This property can be used to ascertain the algorithmic convergence of `FlatNCE`, formalized in the Proposition below. This is significant because non-converging target objective been a big concern in representation optimization [4], and the convergence theory of contrastive learners are currently missing.

E.3 Convergence of FlatNCE

Here we detail the technical conditions for the convergence of FlatNCE to hold. Our derivation follows the analytic framework of generalized SGD from [27], included below for completeness.

Definition S10 (Generalized SGD, Problem 2.1 [27])**.** Let $h(\theta; \omega), \omega \sim p(\omega)$ be an unbiased stochastic gradient estimator for objective $f(\theta)$, $\{\eta_t > 0\}$ is the fixed learning rate schedule, $\{\xi_t > 0\}$ is the random perturbations to the learning rate. We want to solve for $\nabla f(\theta) = 0$ with the iterative scheme $\theta_{t+1} = \theta_t + \tilde{\eta}_t\, h(\theta_t; \omega_t)$, where $\{\omega_t\}$ are iid draws and $\tilde{\eta}_t = \eta_t \xi_t$ is the randomized learning rate.

Assumption S11. (Standard regularity conditions for SGD, Assumption D.1 [27]).

$A1.$ $h(\theta) \triangleq \mathbb{E}_\omega[h(\theta;\omega)]$ is Lipschitz continuous;

$A2.$ The ODE $\dot{\theta} = h(\theta)$ has a unique equilibrium point θ^*, which is globally asymptotically stable;

$A3.$ The sequence $\{\theta_t\}$ is bounded with probability one;

$A4.$ The noise sequence $\{\omega_t\}$ is a martingale difference sequence;

$A5.$ For some finite constants A and B and some norm $\|\cdot\|$ on $\mathbb{R}^d$, $\mathbb{E}[\|\omega_t\|^2] \leq A + B\|\theta_t\|^2$ almost surely $\forall t \geq 1$.

Proposition S12 (Generalized SGD, Proposition 2.2 in [27])**.** Under the standard regularity conditions listed in Assumption S11, we further assume $\sum_t \mathbb{E}[\tilde{\eta}_t] = \infty$ and $\sum_t \mathbb{E}[\tilde{\eta}_t^2] < \infty$. Then $\theta_n \to \theta^*$ with probability one from any initial point θ_0.

Assumption S13. (Weak regularity conditions for generalized SGD, Assumption G.1 in [27]).

$B1.$ The objective function $f(\theta)$ is second-order differentiable;

$B2.$ The objective function $f(\theta)$ has a Lipschitz-continuous gradient, i.e., there exists a constant L satisfying $-LI \preceq \nabla^2 f(\theta) \preceq LI$, where for semi-positive definite matrices A and B, $A \preceq B$ means $v^T A v \leq v^T B v$ for any $v \in \mathbb{R}^d$;

$B3.$ The noise has a bounded variance, i.e., there exists a constant $\sigma > 0$ satisfying $\mathbb{E}\left[\|h(\theta_t;\omega_t) - \nabla f(\theta_t)\|^2\right] \leq \sigma^2$.

Proposition S14 (Weak convergence, Proposition G.2 in [27])**.** Under the technical conditions listed in Assumption S13, the SGD solution $\{\theta_t\}_{t>0}$ updated with generalized Robbins-Monro sequence ($\tilde{\eta}_t$: $\sum_t \mathbb{E}[\tilde{\eta}_t] = \infty$ and $\sum_t \mathbb{E}[\tilde{\eta}_t^2] < \infty$) converges to a stationary point of $f(\theta)$ with probability 1 (equivalently, $\mathbb{E}\left[\|\nabla f(\theta_t)\|^2\right] \to 0$ as $t \to \infty$).

Proposition S15 (Convergence of `FlatNCE`, simple version)**.** Under the technical conditions in Assumption A1, with Algorithm S1 θ_t converges in probability to a stationary point of the unnormalized mutual information estimator $I(\theta) \triangleq \mathbb{E}_{p(x,y)}[g_\theta(x,y)] - \mathbb{E}_{p(x)}[\log Z_\theta(x)]$ (*i.e.*, $\lim_{t\to\infty} \|\nabla I(g_{\theta_t})\| = 0$), where $Z_\theta(x) \triangleq \mathbb{E}_{p(y)}[e^{g_\theta(x,y)}]$. Further assume $I(\theta)$ is convex with respect to θ, then θ_t converges in probability to the global optimum θ^* of $I(\theta)$.

Proof. For fixed $g_\theta(x,y)$ the corresponding optimal $u^*_\theta(x,y)$ maximizing the rhs in Equation (9) is given by

$$u^*_\theta(x,y) = \log \mathbb{E}_{p(y')}[\exp(g_\theta(x,y') - g_\theta(x,y))] \triangleq -\log \mathcal{E}_\theta(x,y), \tag{34}$$

so $\hat{\mathcal{E}}_\theta(x,y) \triangleq \exp^{-\hat{u}_\phi(x,y)}$ can be considered as approximations to $\mathcal{E}_\theta(x,y)$.

$$\begin{aligned} \nabla_\theta\{(9)\} &= -\mathbb{E}_{p(x,y)}\left[e^{-u_\phi(x,y)}\mathbb{E}_{p(y')}[\nabla_\theta \exp(g_\theta(x,y') - g_\theta(x,y))]\right] \qquad (35) \\ &= \mathbb{E}_{p(x,y)}\left[\frac{\hat{\mathcal{E}}_\theta(x,y)}{\mathcal{E}_\theta(x,y)}\nabla_\theta \log \mathcal{E}_\theta(x,y)\right] \qquad (36) \end{aligned}$$

Note $I_{\texttt{BA}} \triangleq \max_{g_\theta}\{\mathbb{E}_{p(x,y)}[\log \mathcal{E}_\theta(x,y)]\}$ is the well-known *Barber-Agakov* (BA) representation of mutual information (*i.e.*, $I_{\texttt{BA}} = I(X;Y)$) [53, 6], so optimizing Equation (9)[8] with SGD is equivalent to optimize $I_{\texttt{BA}}$ with its gradient scaled (randomly) by $\hat{\mathcal{E}}_{\theta_t}/\mathcal{E}_{\theta_t}$ [24].

Under the additional assumption that $\hat{\mathcal{E}}_{\theta_t}/\mathcal{E}_{\theta_t}$ is bounded between $[a,b]$ ($0 < a < b < \infty$), results follow by a direct application of Proposition S12 and Proposition S14. □

[8] Based on the proof of Corollary 3.4, we know `FlatNCE` optimization is a special case of optimizing Equation (9).

F Algorithm for FlatNCE

We summarize the FlatNCE algorithm in Algorithm S1.

Algorithm S1 `FlatNCE`

```
Empirical data distribution $\hat{p}_d = \{(x_i, y_i)\}_{i=1}^{n}$
for $t = 1, 2, \cdots$ do
    Sample $i, i'_k \sim [1, \cdots, n], k \in [1, \cdots, K]$
    $\boldsymbol{g}_\oplus = g_\theta(x_i, y_i), \boldsymbol{g}_\ominus = g_\theta(x_i, y_{i'_k})$
    # $\texttt{logits} = [\boldsymbol{g}_\oplus, \boldsymbol{g}_\ominus], \texttt{labels} = \mathbf{0}$
    # $\ell_{\texttt{InfoNCE}} = \texttt{CrossEntropy}(\texttt{logits}, \texttt{labels})$
    $\texttt{clogits} = \texttt{logsumexp}(\boldsymbol{g}_\ominus - \boldsymbol{g}_\oplus)$
    $\ell_{\texttt{FlatNCE}} = \exp(\texttt{clogits} - \texttt{detach}[\texttt{clogits}])$
    # Use your favorite optimizer
end for
```

G Algorithm for ESS Scheduling

We summarize the effective-sample size (ESS) scheduling scheme in Algorithm S2.

Algorithm S2 ESS Scheduling

```
Empirical data distribution $\hat{p}_d = \{(x_i, y_i)\}_{i=1}^{n}$
Inverse temperature $\beta = 1$, ESS-scheduler $\{\varrho_t \in (1/K, 1]\}_{t=1}^{T}$
Adaptation rate $\gamma = 0.01$
for $t = 1, 2, \cdots, T$ do
    Sample $i, i'_k \sim [1, \cdots, n], k' \in [1, \cdots, K]$
    $\boldsymbol{g}_\oplus = g_\theta(x_i, y_i), \boldsymbol{g}_\ominus = g_\theta(x_i, y_{i'_k})$
    $\texttt{clogits} = \texttt{logsumexp}(\boldsymbol{g}_\ominus - \boldsymbol{g}_\oplus)$
    $\texttt{weights} = \texttt{Softmax}(\boldsymbol{g}_\ominus - \boldsymbol{g}_\oplus)$
    $\texttt{ESS} = 1./(K \cdot \texttt{square}(weights).sum())$
    $\ell_{\texttt{FlatNCE}} = \exp(\texttt{clogits} - \texttt{detach}[\texttt{clogits}])$
    # Use your favorite optimizer
    if $\texttt{ESS} > \varrho_t$ then
        $\beta = (1 - \gamma) \cdot \beta$
    else
        $\beta = (1 + \gamma) \cdot \beta$
    end if
end for
```

H Failed Attempts to Overcome the $\log$-K Curse

The author(s) feel it is imperative to share not only successful stories, but more importantly, those failure experience when exploring new ideas. We contribute this section in the hope it will both help investigators avoid potential pitfalls and inspire new researches.

Joint optimization of primal-dual critics. Inspired by the concurrent research of [24], the author(s) of this paper had originally hope the joint optimization of primal-dual critic as defined in Equation (9) will match, and hopefully surpass the performance of multi-sample `InfoNCE` with single-sample estimation (*i.e.*, $K = 1$). The argument is follows: in theory, the single-sample Fenchel-Legendre estimator has the same expectation with its multi-sample variant, and is provably tighter than `InfoNCE`. In a sense, Fenchel-Legendre estimator is combining the gradient of `FlatNCE` and `InfoNCE`, and the potential synergy is appealing. Unfortunately, in our small scale trial experiments (*i.e.*, MNIST and Cifar), we observe that while the Fenchel-Legendre estimator works reasonably well, it falls slightly below the performance of `InfoNCE` (about 2% loss in top-1 accuracy). We noticed the author(s) of [24] have updated their empirical estimation procedure since first release of the draft, which we haven't experimented with yet on real data. Also our earlier comparison might not be particular fair as we are comparing single-sample versus multi-sample estimators. So this direction still holds promise, which will be investigated in future work.

Alternating the updates of $g(x, y)$ and $u(x, y)$. While our initial attempts with joint optimization of (g, u) failed, we want to use u as a smoothing filtering. This is reminiscent of the exponential moving average trick employed by the MINE, but in a more principled way. Additionally, we further experimented with the idea to optimize on the manifold of u that respects the optimality condition (*i.e.*, $u^*(x, y) = g(x, y) + s(x)$, see [24] for proofs). Contrary to our expectation, these modification destabilizes training. Our estimators exploded after a few epochs, in a way very similar to the DV estimator without sufficient negative samples. The exact reason for this is still under investigation.

I Further Experiments

The above discussion presented several experimental results to highlight unique aspects of the proposetate-of-the-art solutions. Our code can be assessed from https://github.com/Author-name/FlatCLR. All experiments are implemented with PyTorch and executed on NVIDIA V100 GPUs with a maximal level of parallelism at 4 GPUs.

Self-supervised learning (SSL) on Cifar and ImageNet. We set our main theme in SSL and compare the effectiveness of the SimCLR framework [1] to our FlatNCE-powered FlatCLR. Our codebase is modified from a public PyTorch implementation[9]. Specifically, we train 256-dimensional feature representations by maximizing the self-MI between two random views of data, and report the test set classification accuracy using a linear classifier trained to convergence. We report performance based on ResNet-50, and some of the learning dynamics analyses are based on ResNet-18 for reasons of memory constraints. Hyper-parameters are adapted from the original SimCLR paper. For the large-batch scaling experiment, we first grid-search the best learning rate for the base batch-size, then grow the learning rate linearly with batch-size.

The observations made on Cifar align with our theoretical prediction (see Figure 3): in the early training (less than 50 epochs), where the contrast between positives and negatives have not saturated, all models performed similarly. After that, performance start to diverge when entering a regime where FlatNCE learns more efficiently.

We further apply our model to the ImageNet dataset and compare its performance to the SimCLR baseline. We note the SOTA results reported by [1] heavily rely on intensive automated hyper-parameter grid search, and considerably larger networks (*i.e.*, ResNet50 ×4 versus ResNet50), that we are unable to match given our (university-based) computational resources. So instead, we report fair comparison to the best of our affordability. Table 1 reports SSL classification performance comparison up to the 100 epoch[10]. In Table 2 we examine the performance of representation transfer to other datasets. For both cases, FlatCLR consistently outperforms the vanilla SimCLR.

Mini-batch sample MI. In Figure S1 we show that mini-batch sample MI is inadequate for predicting downstream performance.

Large-batch training. In Figure S2 we show large-batch training speedup for the ResNet-50 architecture. Note that we have used the linear scaling of learning rate. And interestingly, for the ResNet-50 architecture model, moderate batch-size (256) actually learned fastest in early training. This implies potential adaptive batch-size strategies to speedup training.

Transfer Learning via a Linear Classifier We trained a logistic regression classifier without l_2 regularization on features extracted from the frozen pretrained network. We used Adam to optimize the softmax cross-entropy objective and we did not apply data augmentation. As preprocessing, all images were resized to 224 pixels along the shorter side using bicubic resampling, after which we took a 224×224 center crop.

[9] https://github.com/sthalles/SimCLR

[10] The reported results is a lower bound to actual performance. We were able to considerably improve the final result via running longer linear evaluation training with larger batch-sizes.

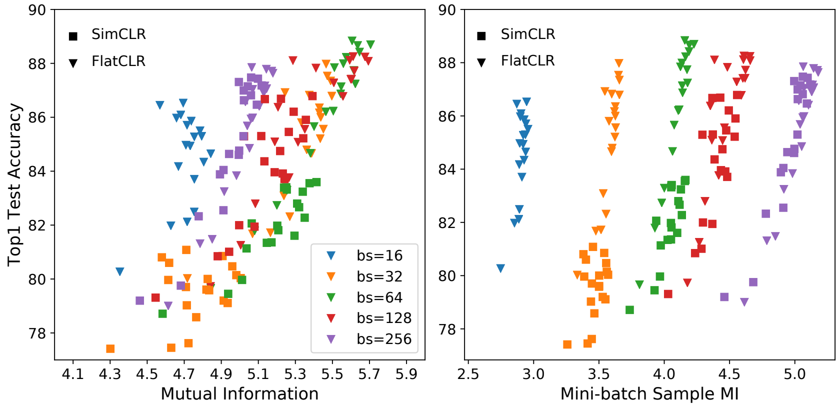

Figure S1: While ground-truth representation MI strongly correlates with performance (left), this relation is not evident with the mini-batch sample MI (right).

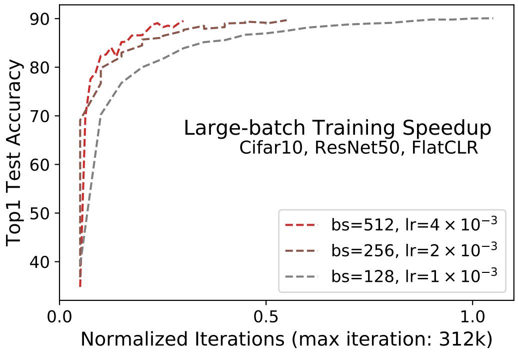

Figure S2: Speed up of large-batch training with ResNet-50 on Cifar. Larger batch leads to faster convergence.

Table S3: ImageNet SSL semi-supervised learning results.

Label fraction	1%		10%	
	Top1	Top5	Top1	Top5
Supervised	5.25	14.40	41.98	67.05
SimCLR	33.44	61.29	54.62	79.89
FlatCLR	**36.35**	**64.59**	**56.51**	**81.32**

Transfer Learning via Fine-Tuning We finetuned the entire network using the weights of the pretrained network as initialization. We trained for 100 epochs at a batch size of 512 using Adam with Nesterov momentum with a momentum parameter of 0.9. At test time, we resized images to 256 pixels along the shorter side and took a 224×224 center crop. We fixed the learning rate = 5^{-5} and no weight decay in all datasets. As data augmentation during fine-tuning, we performed only random crops with resize and flips; in contrast to pretraining, we did not perform color augmentation or blurring.

Semi-supervised Learning Supervised Baselines We compare against architecturally identical ResNet models trained on ImageNet with standard cross-entropy loss. These models are trained with the random crops with resize and flip augmentations and are also trained for 100 epochs.

I.1 Clarifications on the performance gaps to SOTA results

This paper aims for promote a novel contrastive learning objective FlatNCE that overcomes the limitations of the widely employed InfoNCE. While in all experiment we performed, our FlatNCE outperforms InfoNCE under the same settings, we acknowledge that there is still noticeable performance gap compared to SOTA results reported in literature. We want to emphasize this paper is more about bringing theoretical clarification to the problem, rather than beating SOTA solutions, which requires extensive engineering efforts and significant investment in computation, which we do not possess. For example, the SimCLR paper [1] have carried out extensive hyperparameter tuning for each model-dataset combination and select the best hyperparameters on a validation set. The computation resource assessible to us is dwarfed by such need. Their results on transfer learning and semi-supervised learning are transfered from a ResNet50 $(4\times)$ (or ResNet50) with 4096 batch size and 1000 epochs training on SimCLR. Our results posted here are transfered from a ResNet50 with 512 batch size and 100 epochs training on SimCLR and FlatCLR. Also, we chose to use the same hyperparameter and training strategy for each dataset to validate the generalization and present a fair comparison between SimCLR and FlatCLR.

All in all, the author(s) of this paper is absolutely confident that the proposed FlatCLR can help advance SOTA results. We invite the community to achieve this goal together.

J Conclusions

We have presented a novel contrastive learning objective called FlatNCE, that is easy to implement, but delivers strong performance and faster model training. We show that underneath its simple expression, FlatNCE has a solid mathematical grounding, and consistently outperforms its InfoNCE counterpart for the experimental setting we considered. In future work, we seek to verify the effectiveness of FlatNCE on a computation scale not feasible to this study, and apply it to new architectures and applications. Also, we invite the community to find ways to reconcile the performance gap between those theoretically optimal MI bounds and those self-normalized sub-optimal bounds such as FlatNCE and InfoNCE, and develop principled theories for hard-negative training.