Contrastive Representation Learning with Trainable Augmentation Channel

Masanori Koyama, Kentaro Minami, Takeru Miyato
Preferred Networks, Inc.
Tokyo, Japan
{masomatics, minami, miyato}@preferred.jp

Yarin Gal
University of Oxford
Oxford, United Kingdom
yarin@cs.ox.ac.uk

Abstract

In contrastive representation learning, data representation is trained so that it can classify the image instances even when the images are altered by augmentations. However, depending on the datasets, some augmentations can damage the information of the images beyond recognition, and such augmentations can result in collapsed representations. We present a partial solution to this problem by formalizing a stochastic encoding process in which there exist a tug-of-war between the data corruption introduced by the augmentations and the information preserved by the encoder. We show that, with the infoMax objective based on this framework, we can learn a data-dependent distribution of augmentations to avoid the collapse of the representation.

1 Introduction

Contrastive representation learning (CRL) is a family of methods that learns an encoding function h so that, in the encoding space, any set of augmented images produced from a same image (positive samples) are made to attract with each other, while the augmented images of different origins(negative samples) are made to repel from each other [9, 1, 8, 16, 2]. Oftentimes, the augmentations used in CRL are chosen to be those that are believed to maintain the "content"[1] features of the inputs, while altering the "style" features to be possibly discarded in the encoding process [19]. However, how can we be so sure that a heuristically chosen set of augmentations does not affect the features that are important in the downstream tasks? For example, consider applying cropping augmentations T and T' to a dataset consisting of MNIST images located at random position in blank ambient space(Figure1). In this case, since $T'(x_2) = T(x_1)$, training an encoder h such that $h(T'(x_k)) \cong h(T(x_k))$ would also force $h(T(x_1)) \cong h(T(x_2))$ by the transitivity of "$\cong$". In a semi-supervised setting, such a problem of *wrong clustering* may be avoided by considering a stochastic T with a distribution $P(T|X)$ satisfying $P(Y|T(X)) \cong P(Y|X)$, as in [10].

In our study, we provide a partial solution to this problem in a self-supervised setting. In particular, we formalize the representation Z as the output of a stochastic function parametrized by an encoder function h and a stochastic augmentation T, and maximize the mutual information $I(X;Z)$ in a tug-of-war between the data corruption introduced by T and the information preserved by h. Although the infoMax in the context of $I(T(X), T'(X))$ has been discussed in previous literatures [16, 1, 18, 21], it has not been investigated thoroughly while giving a freedom to the distribution of T. We will empirically demonstrate that we can learn a competitive representation by training $P(T|X)$ together with h in this framework. Our formulation of $I(X;Z)$ also provides another way to interpret simCLR [2] as a special case in which $P(T|X)$ is fixed to be the uniform distribution.

[1] If Y is a target signal, we may for example assume $P(Y|T(X)) = P(Y|X)$, as in [10]

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

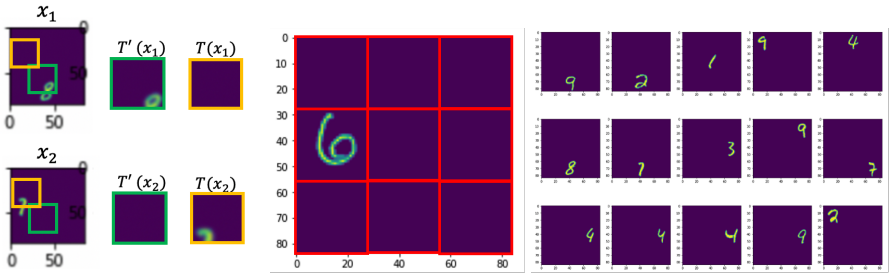

Figure 1: The leftmost Panel: If we enforce the equivalence relation $T(x_k) \sim T'(x_k)$, then we will also have $T(x_2) \sim T(x_1)$ by transitivity because $T'(x_2) = T(x_1)$. Right two panels: Example images of the MNIST-derived dataset and 9 positions on which a MNIST digit was placed in each one of $(28 * 3) \times (28 * 3)$ dimensional image.

2 InfoMax problem with Augmentataion Channel

Existing perspectives of CLR are based on $I(T(X); T'(X))$, the mutual information between $T(X)$ and $T'(X)$, where T and T' are equally distributed (independent) random augmentations (discussed more in depth in related works, Section 4). In this work, we revisit the infoMax problem from a different perspective in a framework of self-supervised learning that explicitly separates the *augmentation channel* in the encoding map $X \to Z$. Consider the generation process illustrated in the Figure 2. In this process, $V = T(X)$ is produced from X by applying a random augmenation T

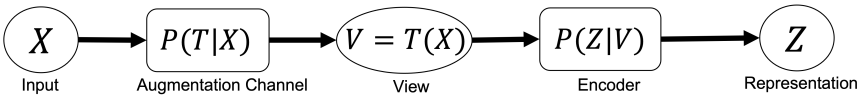

Figure 2: Generation Process of Z

sampled from some distribution $P(T|X)$. V is then encoded into Z through the distribution $P(Z|V)$ parametrized by some encoder h. Thus, the distribution of Z can be written as

$$P(z|x) = \int P(z|T(x))\, P(T|x) dT \tag{1}$$

where we use lowercase letters to denote the realization of random variable (e.g. x is a realization of X.) . Using $\mathcal{G}$ to denote the family of distributions that can be written in this form, we consider the InfoMax problem $\max_{p \in \mathcal{G}} I(X; Z)$. In this definition of the map $X \to Z$, the support $\mathcal{T}$ of $P(T|X)$ determines the maximum amount of information that can be preserved. For example, if all members of $\mathcal{T}$ strongly corrupts X, $I(X; Z)$ would be small for all choice of $P(T|X)$. Meanwhile, if the identity transformation is included in $\mathcal{T}$, then $V = X$ can be achieved by setting $P(T|X) = \delta_{id}(T)$. However, as in training methods based on noise regularization [11, 14, 10], the identity mapping is often not included in the augmentation set because it does not help regularize the model.

The infoMax problem in our framework has a deep connection with modern self supervised learning, as it can provide another derivation of simCLR that does not use a variational approximation.

Proposition 1. Suppose that $P(Z \mid T(X)) = C_\beta \exp(\beta \mathcal{S}(Z, h(T(X))))$ where $\mathcal{S} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is a similarity function on the range of Z and C_β is a constant dependent only on β. Then

$$I(X; Z) = E_{X,Z}\left[\log E_{T'|X}\left[\frac{\exp(\beta \mathcal{S}(Z, h(T'(X))))}{E_{T'', \tilde{X}}[\exp(\beta \mathcal{S}(Z, h(T''(\tilde{X}))))]}\right]\right] \tag{2}$$

Also, when $P(T|X)$ is uniformly distributed on a compact set of view-transformations, the mean approximation of Z and Jensen's inequality on the $E_{T'|X}$ part of (2) recovers the simCLR loss.

For the proof of Prop 1, please see Appendix 5.1. We shall note that the condition of this statement is fulfilled in natural cases, such as when $P(Z|T(X))$ is Gaussian or Gaussian on the sphere. In the proof of Prop 1, the numerator and the denominator correspond directly to the entropies $-H(Z|X)$

Table 1: Linear evaluation accuracy scores. Raw Representation achieves 0.8992 ± 0.0012. For the description of *oracle* and *topn*, please see the main script (Section 3.1). While it is somewhat obvious that vanilla simCLR fails in our experimental setting, it is an important indication that random augmentations with arbitrary distribution can harm the task performance.

Method	Ours	Ours(topn)	SimCLR	simCLR(oracle)
Projection Head	0.95505 ± 0.0023	0.9552 ± 0.0037	0.3156 ± 0.0044	0.5144 ± 0.011
f output	0.9729 ± 0.0014	0.9748 ± 0.0012	0.4598 ± 0.0056	0.9354 ± 0.0029

and $H(Z)$. If Z takes its value on the sphere $\mathcal{S}^d$, enlarging $H(Z)$ would encourage Z to be uniformly distributed over the sphere. These observations support the theory proposed in [20]. The table in Appendix 5.2 summarizes our algorithm for optimizing the objective (2) with respect to both $P(T|X)$ and h.

3 Experiments

We show that, by training $P(T|X)$ together with the encoder h based on the objective (2), we can learn a better representation than the original simCLR. We conducted an experiment on a dataset derived from MNIST mentioned at the introduction (Figure1). To construct this dataset, we first prepared a blank image of size $(28 * 3) \times (28 * 3)$, which is 3 times greater in both dimensions than the original MNIST images (28×28). We then created our dataset by placing each MNIST image randomly at one of $3 \times 3 = 9$ grid locations in the aforementioned blank image. We set T to be a random augmentation that crops a 20×20 image at one of $17 \times 17 = 289$ locations ranging over the $(28 * 3) \times (28 * 3)$ dimensional image with stride size 4. That is, T is a categorical random variable with 289 categories and $P(T|X)$ is a softmax function whose output is constructed from a 289 dimensional vector. On this dataset, any crop that does not intersect with the digit produces the same *empty* image, which is useless in discriminating the image instances. For computational ease, we trained our encoder h based on the Jensen-lower bound of (2). We shall also note that, in our setup, our h corresponds to the composition of the projection head g and the encoder f in the context of the recent works of contrastive learning. We evaluated the representation of both $h = g \circ f$ and f. Also, without any additional constraint, $P(T|X)$ sometimes collapsed to the "the most discriminating" crop on the training set, resulting in a representation that does not generalize on the downstream classification task. To resolve this problem, we adopted the maximum entropy principle [6] and optimized our objective (2) together with small entropy regularization $H(T|X)$, seeking the highest entropy T that maximises the objective (2).

3.1 Performance of the trained representations in Linear Evaluation Protocol

To evaluate the learned representation, we followed the linear evaluation protocol as in [2] and trained a multinomial logistic regression classifier on the features extracted from the frozen pre-trained network. We used Sklearn library [12] to train the classifier. For SimCLR, it is often customary to use the "center crop" augmentation T_{center} and report $h(T_{center}(X))$ as the representation for X. However, in this example, "center crop" would extract an empty image with high probability. Thus, we computed the representation of each X by integrating the encoded variable with respect to $P(T|X)$, that is, $\hat{Z} = E_{T \sim P(T|X)}[h(T(X))]$ ($P(T|X)$ for simCLR is uniform). For the models with non-uniform $P(T|X)$ we also evaluated Z_{topn}, the representation obtained by averaging $h(T(X))$ over the set of Ts having the top eight $P(T|X)$ density. As an ablation, we also evaluated the SimCLR-trained encoder by integrating its output with respect to the oracle $P(T|X)$ concentrated uniformly on the 9 crop positions with maximal intersection with the embedded MNIST image. We conducted each experiment with 4 seeds. The table 1 summarizes the result.

We can see that, with our trained $P(T|X)$ and h, we can achieve a very high linear evaluation score, even better than the raw representation result on the ordinary MNIST dataset (0.9256). Interestingly, with our $P(T|X)$, the representation is competitive even at the projection head, and its performance even exceeds the representation of simCLR obtained by averaging $f(T(X))$ over the oracle $P(T|X)$. This trend was also observed in the experiment on the original MNIST(see Appendix 5.4). This result may suggest that the poor quality of simCLR representation at the level of the projection head is partially due to the fact that proper $P(T|X)$ is not used in training the model. Also, in confirmation

of our problem statement in the section 1, the representation learned without the trainable $P(T|X)$ collapses around that of the empty image (see Appendix 5.5). In terms of the average pairwise Gaussian potential used in [20] that measures the uniformity of the representations on the sphere(lower the better), our representation achieves 0.0845 as opposed to 0.9757 of the baseline simCLR.

3.2 The trained $P(T|X)$ agrees with our intuition

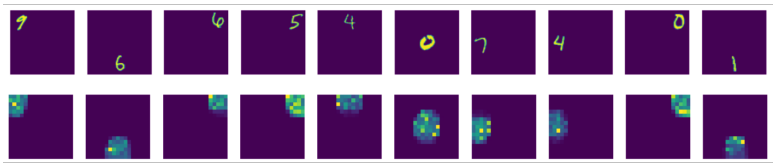

Figure 3: Visualization of the trained $P(T|x)$ (bottom row) for various choice of x(top row). Brighter color represents higher intensity.

Figure 3 visualizes the density of $P(T|x)$(second row) for various input image x(second row). In each image of the second row, the intensity at (i, j)th pixel is $P(T_{ij}|X)$, where T_{ij} is the augmentation that crops the sub-image of size 20×20 with the top left corner located at (i, j). As we can see in the figure, the learned $P(T|X)$ is concentrated on the place of digit, ignoring the crop locations that would return the empty image. Our learned $P(T|X)$ in fact captures the non-trivial crop with probability 0.998 ± 0.003 on 10,000 test images.

4 Related Works and conclusion

In a way, $P(T|X)$ can be considered an augmentation *policy*. [3, 7] also learns $P(T|X)$ with supervision signals. [13] extends these works to self-supervised setting by applying a modified [7] to a set of self-supervised tasks that are empirically correlated to the target downstream tasks.

There also are several works that investigate the importance of non-uniform sampling in the constrative learning. For example, [17] proposes the infoMin principle, which claims that one shall engineer the distribution of $T(X)$ in such a way that it (1) shares as much information as possible with the target variable Y while (2) ensuring that, for any two realization $t_1 \neq t_2$ of T, $t_1(X)$ and $t_2(X)$ should have as little information in common. In their work, however, they do not provide an algorithm to optimize the distribution of T. In a way, the requirements (1) and (2) seem to be respectively related to $H(X|Z)$ and $H(Z)$ in the numerator-denominator decomposition of (2). Also, because they are practically conducting an empirical study on the joint distribution $P(T_1, T_2)$, their work might be also related to the optimization of $P(Z|T(X))$ in our context. Also, [15] trains T adversarially with respect to the loss. However, in the setting we discuss in this paper, this strategy would encourage T to crop only the empty image and collapse the representations.

Previously, the connection between CLR and Mutual information has also been described based on the perspective that interprets CLR as a variational approximation of the mutual information between two views $I(V_1; V_2)$, where each $V_k = T_k(X)$ is a "view" of X produced by some augmentation function T_k [16, 1, 18, 21]. This variational approximation is based on the inequality

$$I(V_1; V_2) \geq E_{V_1,V_2}\left[\frac{\exp(f(V_1, V_2))}{E_{V_1'}[\exp(f(V_1', V_2)]}\right] \tag{3}$$

that holds for *any* measurable f. Based on this infoNCE perspective, [18] considers a case in which Z is trained as $Z = g(V)$ with invertible g, and presents an empirical study suggesting that simCLR can improve the representation even in this setting. Based on this argument, [18] suggests that $I(Z_1, Z_2)$ cannot be used to explain the success of simCLR. However, as we point in our study, the transformation $X \to V$ usually involves information *loss* via augmentations like *cropping*, and CLR is often evaluated based on Z sampled from $P(Z|V)$. In this study, we formalize the augmentation channel $X \to V$ as a part of $X \to Z$, and present a result suggesting that, at least for the learning of $P(T|X)$, the Mutual information (MI) with $H(T|X)$ regularization might be an empirically useful measure for learning a good representation, in particular at the level of final ouput(projection head). Our result may suggest that it might be still early to throw away the idea of MI in all aspects of the CLR because [18] studies a case in which only the $V \to Z$ part of $X \to Z$ is made invertible.

It might also be worthwhile to mention some theoretical advantages of our formulation. Because (3) is a variational bound that holds for any choice of f, this inequality does not help in estimating how much the RHS derived from a *specific* choice of f (i.e. RHS(f)) differs from $I(V_1; V_2)$. Also, when we optimize RHS(f) using a popular family of f defined as $f(V_1, V_2) := \psi(h(V_1))^T \psi(h(V_2))$ [21], there is no way to know "in what proportion a given update of f would affect $I(h(V_1); h(V_2))$ and $I(V_1; V_2) - RHS(f)$. Meanwhile, in our formulation, the difference between simCLR and MI is described directly with Jensen and mean approximation, for which there are known mathemtical tools like [5]. It might be interesting to further investigate the claims made by [18] in this direction as well. We believe that our approach provides a new perspective to the study of contrastive learning as well as insights to the choice of augmentations.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in Neural Information Processing Systems(NeurIPS)*, 2019.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *International Conference on Machine Learning(ICML)*, 2020.

[3] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2019.

[4] Rick Durrett. *Probability: Theory and Examples*. Brooks/Cole Thomson, 2019.

[5] Xiang Gao, Meera Sithram, and Ardian E. Roitberg. Bounds on the jensen gap, and implications for mean concentrated distributions. *The Australian Journal of Mathematical Analysis and Applications*, 16, 2016.

[6] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *International Conference on Machine Learning(ICML)*, 2017.

[7] Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation. *European Conference on Computer Vision(ECCV)*, 2020.

[8] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *International Conference on Machine Learning(ICML)*, 2020.

[9] 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. *International Conference on Learning Representations(ICLR)*, 2019.

[10] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. *International Conference on Machine Learning(ICML)*, 2017.

[11] Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, 2018.

[12] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research(JMLR)*, 12(Oct): 2825–2830, 2011.

[13] Colorado J Reed, Sean Metzger, Aravind Srinivas, Trevor Darrell, and Kurt Keutzer. Selfaugment: Automatic augmentation policies for self-supervised learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, 2021.

[14] Jonas Rothfuss, Fabio Ferreira, Simon Boehm, Simon Walther, and Andreas Krause Maxim Ulrich, Tamim Asfour. Noise regularization for conditional density estimation. *arXiv preprint arXiv:1907.08982*, 2019.

[15] Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning. *International Conference on Learning Represenations(ICLR)*, 2021.

[16] Yonglong Tian, Dilip Krishna, and Phillip Isola. Contrastive multiview coding. *European Conference on Computer Vision(ECCV)*, 2019.

[17] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, and Philip Isola Cordelia Schmid. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems(NeurIPS)*, 2020.

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

[19] Julius von KÃijgelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard SchÃűlkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *arXiv preprint arXiv;2106.04619*, 2021.

[20] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *International Conference on Machine Learning(ICML)*, 2020.

[21] Mike Wu, Chengxu Zhuang, Milan Mosse, Daniel Yamins, and Noah Goodman. On mutual information in contrastive learning for visual representations. *arXiv preprint arXiv:2005.13149*, 2020.

5 Appendix

5.1 Formal statement and the proof of Proposition 1

Prposition. Suppose that $p(Z \mid T(X)) = C_\beta \exp(\beta \mathcal{S}(Z, h(T(X))))$ where $\mathcal{S} : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is a similarity function and C_β is a normalization constant dependent only on β, Then

$$I(X;Z) = E_{X,Z}\left[\log E_{T'|X}\left[\frac{\exp(\beta \mathcal{S}(Z, h(T'(X))))}{E_{T'',X'}[\exp(\beta \mathcal{S}(Z, h(T''(X'))))]}\right]\right]. \tag{4}$$

Also, when $P(T|X)$ is uniformly distributed over a compact set of view-transformations, we recover the loss of SimCLR by (1) applying Jensen's inequality on $E_{T'|X}$ and (2) approximating Z with $h(T(X))$, the mean of $p(Z|T(X))$.

Proof. We use upper case letter to denote the random variable and lower case letter to denote its corresponding realization (x is a realization of X). We also use the standard notation in the measure theoretic probability that treat expressions like $P(A|B)$ and $E[A|B] := E_{A|B}[A]$ as a random variable that is measurable with respect to B. Thus, in the equality $E[A] = E[E[A|B]]$, the integral $E[A|B]$ inside the RHS is a random variable with respect to B. To clarify, we sometimes use the subscript to represent the variable with respect to which the integral is taken. For more details about this algebra, see [4] for example. Here, we show the proof of the version of the statement with the application of Jensen's inequality. The proof without Jensen's inequality can be derived easily from the intermediate results of this proof.

On $-H(Z \mid X)$

$$\begin{aligned}
E_{X,Z}[\log P(Z|X)] &= E_{X,Z}[\log(E_{T'}[P(Z|X,T')|X])] && (5)\\
&:= E_{X,Z}[\log E_{T'|X}[(C_\beta \exp(\beta S(Z, h(T'(X)))))]] && (6)\\
&= E_{X,Z}[\log E_{T'|X}[(\exp(\beta S(Z, h(T'(X)))))] + C_\beta && (7)\\
&\geq E_{X,Z}[E_{T'|X}[\log(\exp(\beta S(Z, h(T'(X)))))] + C_\beta && (8)\\
&= E_{X,Z}[E_{T'|X}[\beta S(Z, h(T'(X)))] + C_\beta && (9)\\
&:= E_{X,Z}[E_{T'|X}[\beta S(Z, h(T'(X)))] + C_\beta && (10)
\end{aligned}$$

On $H(Z)$

$$\begin{aligned}
-E[\log P(Z)] &= -E_Z[\log(E_{X',T''}[P(Z|X',T'')])] && (11)\\
&= -E_Z[\log(E_{X',T''}[C_\beta \exp(\beta S(Z, h(T''(X'))))])] && (12)\\
&= -E_Z[\log(E_{X',T''}[\exp(\beta S(Z, h(T''(X'))))])] - C_\beta && (13)
\end{aligned}$$

Altogether, we see that C_β cancels out and

$$\begin{aligned}
H(Z) - H(Z \mid X) &\geq E_{X,Z}[E_{T'|X}[\beta S(Z, h(T'(X)))] + C_\beta && (14)\\
&\quad - \log(E_{X',T''}[exp(\beta S(Z, h(T''(X'))))])] - C_\beta && (15)\\
&= E_{X,Z}\left[E_{T'|X}\left[\log \frac{\exp(\beta S(Z, h(T'(X))))}{E_{X',T''}[\exp(\beta S(Z, h(T'(X'))))]}\right]\right] && (16)
\end{aligned}$$

The equality emerges if we do not apply Jensen's inequality on $-H(Z|X)$.

To show the connection of this result with simCLR, we approximate $Z|X$ as $h(T(X))$, the mean of $P(Z|T(X))$. With this approximation, the outermost integration with respect to (X, Z) will be replaced by the integration with respect to (X, T). Also, because T'' is integrated away in the

denominator of (16), the *double prime* superscript of the T'' is superficial. Thus, we obtain

$$E_{X,T}\left[E_{T'|X}\left[\log \frac{\exp(\beta S(h(T(X)),h(T'(X))))}{E_{X',T'}[\exp(\beta S(h(T(X)),h(T'(X'))))]}\right]\right] \tag{17}$$

$$\cong \frac{1}{N}\sum_{x_i\sim X, T_i\sim(T|x_i)}\left(\frac{1}{\tilde{N}}\sum_{T'_k\sim(T|x_i)}\beta S(h(T_i(x_i)),h(T'_k(x_i)))\right) \tag{18}$$

$$-\frac{1}{M}\log\left(\sum_{x_j\sim X, T'_j\sim(T|x_j)}\exp\left(\beta S(h(T_i(x_i)),h(T'_j(x_j)))\right)\right)\Bigg) \tag{19}$$

With $i \in 1:N, k \in 1:\tilde{N}, j \in 1:M$.

Choosing $\tilde{N}=1$ and $M=N$, we get

$$\frac{1}{N}\sum_{x_i\sim X, T\sim(T|x_i), T'_i\sim(T|x_i)}\log\left(\frac{\exp(\beta S(h(T_i(x_i)),h(T'_i(X_i)))}{\frac{1}{N}\log\left(\sum_{x_j\sim X, T'_j\sim(T|x_j)}\exp\left(\beta S(h(T_i(x_i)),h(T'_j(x_j)))\right)\right)}\right) \tag{20}$$

which agrees with the simCLR loss when $T|X$ is set to be uniform.

□

5.2 Algorithm

The table shown below is the description of the algorithm based on Proposition 1 that trains h and $P(T|X)$ together. In this algorithm we assume that the support of $P(T|X)$ is discrete. Instead of training h and $P(T|X)$ simultaneously, we train h and $P(T|X)$ in turn because this strategy was able to produce more stable results. With this algorithm's notation, the very classic SimCLR would emerge if we set m(the number of T samples) to be 2 and set $P(T|X)$ to be uniform. In our experiments we set m to be 8, as it performed better than anything less for both fixed $P(T|X)$(SimCLR) and trainable $P(T|X)$.

Algorithm 1 Contrastive Representation learning with trainable augmentation Channel(CRL-TAC)

Require: A batch of samples $\{x_k\}$, an encoder model $h_\theta : x \to z$, the number of transformation samples m, a model for conditional random augmentation distribution $x \to P(T|x,\eta)$

1: **for** each iteration i **do**
2: **Update phase for** h
3: Sample $T_{jk} \sim P(T|x_k,\eta)$, $j = 1,...,m$
4: Apply $\{T_{jk}; j = 1,...,m\}$ to each x_k, producing a total of $m \times k$ samples of $T_{jk}(x_k)$.
5: Empirically compute the objective (2) or its lower bound, and update θ
6: **Update phase for** $P(T|X)$
7: Sample $T_{jk} \sim Uniform$
8: Evaluate (2) with $P(t_j|x_k,\eta)$ weights, and update η
9: **end for**

5.3 Model Architecture and entropy regularization

In our experiment, we used a three layer CNN with 200 dimensional output for the intermediate encoder f and a two layer MLP with 50 dimensional output for the projection head g(Figure 4). We chose this architecture because this choice performed stably for SimCLR on standard MNIST dataset (See Section 5.4). We trained $P(T|X)$ with three layer CNN(Figure 5).

As in [20], we normalized the final output of the encoder $h = f \circ g$ so that the final output is distributed on the sphere. As such, we used $S(a,b) = a^T b$, and set $\beta = 0.5$ since this choice yielded stable results for the learning of $P(T|X)$. At the inference time, we normalized $E_{P(T|X)}[h((T(X))]$. To discourage $P(T|X)$ from collapsing prematurely, we imposed a regularization of $H(T|X)$ with

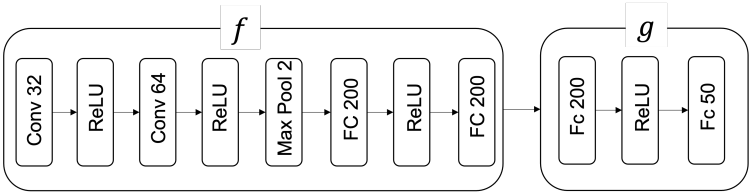

Figure 4: Encoder architecture

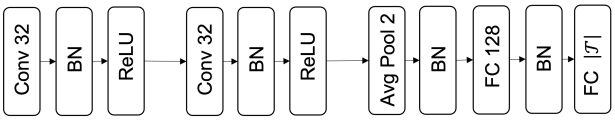

Figure 5: $P(T|X)$ architecture

coefficient λ. We used coefficient $\lambda = 0.0025$, as it achieved the lowest contrastive loss on the training set in the range $[0.001, 0.005, 0.0025]$.

This choice of λ also produced the best linear evaluation score on the training dataset. Setting $\lambda < 0.0001$ seemed to collapse $P(T|X)$ in many cases.

5.4 Results on the original MNIST dataset

Table 2 shows the results on the original MNIST dataset. We used the same setting as for the main experiment in Section 3, except that we set $\beta = 1.0$. On this dataset, raw representation achieves 0.9255. When trained with uniform $P(T|X)$, the projection head representation is not much better than the raw representation. However, when trained together with $P(T|X)$, the projection head representation is comparable to the f output. This result also suggest that, by training $P(T|X)$ together with $h = g \circ f$, we can improve the utility of the representation at the level on which the objective function function is trained, instead of the heuristically chosen intermediate representation f. This result also suggests that there is much room left for the study of the stochastic augmentation and intermediate representation.

Table 2: Linear evaluation accuracy Scores on the original MNIST dataset. Raw Representation achieves 0.9255 ± 0.0001 on the original MNIST dataset.

Method	ours	ours(topn)	SimCLR
Projection Head	0.9642 ± 0.0025	0.9674 ± 0.0015	0.9273 ± 0.0044
f output	0.9805 ± 0.0006	0.9859 ± 0.0004	0.9806 ± 0.0056

5.5 Uniformity of the learned representation

[20] reports that, for a good representation, the representation tends to be more uniformly distributed on the sphere. The graphs in Figure 6 are scatter plots of 2-dimensional representations trained with and without the trainable $P(T|X)$. The graphs in Figure 7 are superimposed plots of 50 dimensional representations with and without the trainable $P(T|X)$. On these graphs, we can visually see that what we feared in Section 1 and Figure 1 happens when we fix $P(T|X)$; the majority of the representations becomes strongly concentrated around that of the empty image. This problem is successfully avoided with the trainable $P(T|X)$. In terms of the average pairwise Gaussian potential used in [20] that measures the uniformity of the representations on the sphere(lower the better), our 50 dimensional representation achieves 0.0845 as opposed to 0.9757 of the baseline SimCLR with fixed $P(T|X)$. The graphs in Figure 8 are the sorted values of $|\langle h(x), h(x')\rangle|$ for a randomly sampled set of (x, x') pairs. We see in these graphs that the representations with the trainable $P(T|X)$ are trained to be as orthogonal to each other as possible($|\langle h(x), h(x')\rangle|$ is concentrated around 0) , while the representations trained with the fixed $P(T|X)$ are collapsing into one direction ($|\langle h(x), h(x')\rangle|$ is concentrated around 1).

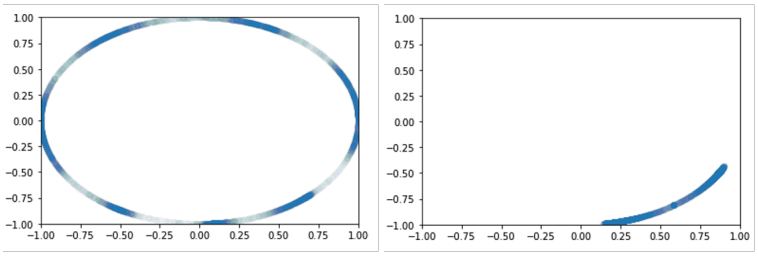

Figure 6: Left: The scatter plot of 2 dimensional representations trained together with $P(T|X)$. Right: The scatter plot of 2 dimensional representations trained with uniform $P(T|X)$.

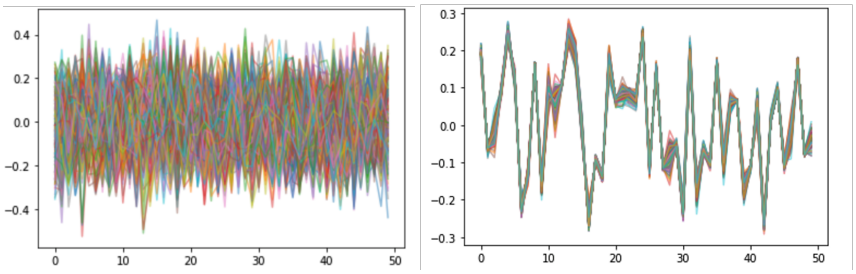

Figure 7: Left: The superimposed plot of randomly sampled 200 instances of 50 dimensional representations trained together with $P(T|X)$. The horizontal axis represents the indices of the vectors, and each curve with a different color represents one instance of the vector $h(x) \in \mathcal{R}^{50}$. Right: The superimposed plot of 50 dimensional representations trained with uniform $P(T|X)$. We see that all instances of $h(x)$ look very similar.

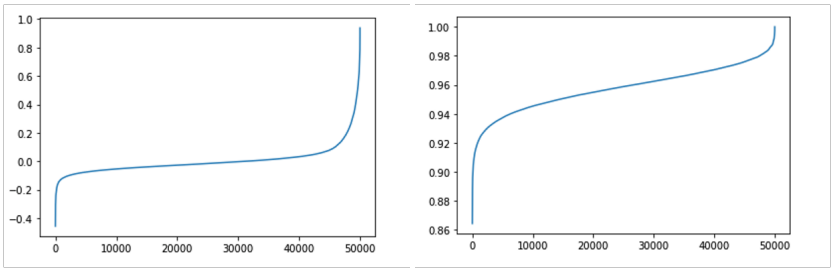

Figure 8: Left : The plot of the sorted values of $|\langle h(x), h(x)\rangle|$ for a randomly sampled sets of (x, x') pairs, when each $h(x)$ is a 50 dimensional representation trained together with $P(T|X)$. Right: The same figure with h trained with uniform $P(T|X)$.