On the Importance of Views in Unsupervised Representation Learning

Mike Wu[†], Chengxu Zhuang[‡], Daniel Yamins[†‡], Noah Goodman[†‡]
Department of Computer Science[†] and Psychology[‡]
Stanford University
{wumike, chengxuz, yamins, ngoodman}@stanford.edu

January 23, 2020

Abstract

In recent years, several self-supervised algorithms in vision have been shown to learn rich representations that perform remarkably well on downstream transfer tasks. We show that these state-of-the-art visual representation learning methods maximize an objective function that is a lower bound on the mutual information between two or more "views", where a view is defined by a composition of image augmentations. This formulation provides a unified theory that connects these vision models (e.g. Instance Discrimination, Local Aggregation) with well-known language representation learning methods like BERT. To do this, we introduce a new lower bound on mutual information inspired by InfoNCE that we prove supports drawing negative samples from a restricted class of variational distributions. Using the mutual information perspective, we find that the choice of (1) negative samples and (2) "views" are critical to the success of this family of algorithms.

1 Introduction

While large supervised networks have given rise to human-level performance in several visual [21, ?, 8] and language-based tasks [23, ?, 3], they require an exhaustive amount of labelled data, posing a barrier to widespread adoption. In recent years, we have seen the growth of several approaches to *un*-supervised learning in both the vision and natural language communities [18, 3, 15, 25, 10, 24, 26, 7]. Here, the desiderata is learning semantic representations (without labels) that are simultaneously useful for several downstream tasks. Common transfer tasks include visual question answering, image classification, or the GLUE benchmarks [23, 22] to name a few.

There are seemingly a number of distinct recipes for learning a good representation. In language, training paradigms span several cleverly designed self-supervised objectives. For instance, Skip-Gram [16] and Skip-Thought [12] learn a representation that is predictive of the words or sentences around it in natural text. More recently, BERT [3], XLNet [25], and RoBERTa [15] rely on predicting randomly masked tokens. In vision however, unsupervised methods (e.g. Instance Discrimination [24] and Local Aggregation [26]) focus on discriminating between individual images with the intuition that a better representation can pick out any instance from a large set. On the surface, the two groups appear unrelated. In particular, the vision objectives are *external*, in the sense that they focus on the relation between many data points whereas the language objectives are *internal*, focusing on the relations between parts of a single example.

In this paper, we hope to bridge algorithms under a single unifying view of mutual information. Kong et. al. [13] has already shown that many of the language methods (e.g. BERT) maximize a lower bound on the mutual information between disjoint "views" of a sentence. For example, a sentence with masked tokens and the masked tokens themselves comprise two "views" in the BERT paradigm. Here, we further show that the visual clustering objectives [24, 26] also maximize a lower bound on mutual information between "views" of an image, where a "view" is defined by the composition of image augmentations. By intepreting these algorithms under a unified theory, we hope to motivate many potentially interesting directions for future representation learning algorithms.

Concisely, our contributions are as follows:

- We provide conditions and a formal proof missing from previous literature that InfoNCE with negative samples drawn from a restricted family of *variational distributions* lower bounds mutual information. We call this estimator Variational InfoNCE (VINCE).
- We frame several popular unsupervised algorithms (masked language models, instance discrimination, local aggregation) as VINCE estimators.
- A "view" is currently a very vague notion. We provide a definition and properities of a good "view".
- We find two critical choices common to these unsupervised algorithms: (1) how to choose negative samples, and (2) how to choose a set of "views" of each example. Through experiments, we find that (1) and (2) are the *crux* of the success of the mutual information objective.

2 What is Mutual Information?

Mutual information, denoted by $\mathcal{I}(X;Y)$, measures the statistical dependence between two random variables, X and Y. Intuitively, $\mathcal{I}(X;Y)$ can be interpreted as how much observing the value of Y reduces uncertainty of the value of X. For instance, two independent random variables have a mutual information of 0 (knowing one provides no new information about the other). On the other extreme, if X is a deterministic function of Y alone, mutual information is equivalent to the entropy of Y (knowing one pinpoints the distribution of the other). Formally, we write mutual information as

$$\mathcal{I}(X;Y) = \mathbb{E}_{p(x,y)}\left[\log \frac{p(x,y)}{p(x)p(y)}\right] = \mathcal{I}(Y;X) \tag{1}$$

where $p(x,y)$ is the true joint distribution over X and Y. From the middle expression, we can view mutual information as the Kullback-Liebner (KL) divergence between the joint and the product of the marginals.

3 Bounds on Mutual Information

In general, mutual information is computationally intractable to compute. This is especially true in machine learning settings as X and Y are frequently high dimensional random variables with infinite support. To circumvent infeasible integrals, there have been several approaches to lower bound mutual information [20], of which a popular one is noise-contrastive estimation [6], also called InfoNCE [19]. The InfoNCE estimator is written as[1]

$$\mathcal{I}^{\text{NCE}}(X;Y) = \mathbb{E}_{p(x,y_1)}\left[f_{\theta,\phi}(x,y_1) - \mathbb{E}_{p(y_{2:K})}\left[\log \frac{1}{K}\sum_{i=1}^{K} e^{f_{\theta,\phi}(x,y_i)}\right]\right] \tag{2}$$

where $f_{\theta,\phi}(x,y) = g_\theta(x)^T g_\phi(y)$ be a *witness* function representing the "compatibility" of two values. We use g_θ and g_ϕ to designate encoders that map realizations of X and Y to vectors. We call the output of each encoder a *representation.* In practice, ϕ and θ are chosen via stochastic gradient descent (SGD) to maximize $\mathcal{I}^{\text{NCE}}(X,Y)$, finding the "optimal" witness. The value of the witness function alone is an unnormalized quantity. Thus, given a particular x and y, the second term in Eq. 2 serves to normalize $f_{\theta,\phi}(x,y)$ with respect to other plausible realizations of Y. We use $y_{2:K} = \{y_2,\ldots,y_K\}$ to denote a set of K *negative samples* used to estimate the marginal probability. A common choice for the distribution over sets is $p(y_{2:K}) = \prod_{i=2}^{K} p(y_i)$, in other words i.i.d. samples from the marginal $p(y)$. In Sec. A, we include a proof to show $\mathcal{I}(X;Y) \geq \mathcal{I}^{\text{NCE}}(X;Y)$ largely based on [20].

One of the important choices we will frequently revisit is where negative samples are drawn from. InfoNCE samples independently from $p(y)$ but this choice may not always be desirable. Previous papers [20] have found that InfoNCE requires a large number of negative samples to reduce the variance of its estimate. We might, for example, want to choose a different "proposal" distribution over sets $q(y_{2:K})$ that is more efficient in estimating the marginal. We may even wish to conditionally sample negatives, $q(y_{2:K}|y_1)$ where $y_1 \sim p(y_1)$. While previous literature presents InfoNCE with an arbitrary variational distribution [19, 13] that would justify either of these choices, we have not found a proof supporting this. One of the contributions of this paper is to formally define a

[1]You might see the InfoNCE estimator written without the $\frac{1}{K}$ in literature. As $-\log K$ is a negligible constant in optimization, the two forms are proportional.

class of variational distributions $q(y_{2:K}|y_1)$ such that Eq. 2 remains a valid lower bound if we replace $p(y_{2:K})$ with $q(y_{2:K}|y_1)$.

Theorem 3.1. *Let X, Y_1 be two random variables. Let $p(y_1)$ be the marginal distribution for Y_1. Given a sample $y_1 \sim p(y_1)$, define conditional distributions $q(y_i|y_1)$, $i = 2$ to K with the same support as $p(y_1)$ from which we can easily sample from (but not necessarily score). Consequently, define $Y_2, \ldots Y_K$ as auxiliary random variables distributed according to each respective $q(y_i|y_1)$. The Variational InfoNCE estimator (VINCE) is*

$$\mathcal{I}^{\text{VINCE}}(X;Y_1) = \mathbb{E}_{p(x,y_1)q(y_{2:K}|y_1)}\left[\prod_{i=2}^{K}\frac{p(y_i)}{q(y_i|y_1)}\left(\log\frac{e^{f_{\theta,\phi}(x,y_1)}}{\sum_{j=1}^{K}e^{f_{\theta,\phi}(x,y_j)}}\right)\right] - K + 1 \tag{3}$$

where $q(y_{2:K}|y_1) = \prod_{j=2}^{K} q(y_i|y_1)$. Then, $\mathcal{I}(X;Y_1) \geq \mathcal{I}^{\text{VINCE}}(X;Y_1)$.

Proof. See Sec. B in Appendix. □

However, Eq. 3.1 is somewhat meaningless: we cannot evaluate $p(y)$ nor $q(y|y_1)$. In the following Corollaries, we add two different restrictions on $q(y|y_1)$ that define two families of tractable variational distributions. For each family, we provide examples of members.

Corollary 3.1.1. *Fix $y_1 \sim p(y_1)$. Define $q(y_i|y_1)$ to be the probability density induced by rejection sampling (with any acceptance strategy) on samples from $p(y_i)$. Specifically, set $q(y_i|y_1) = r(y_i|y_1)p(y_j)$ where $r(y_i|y_1) \in (0,1]$ is an acceptance probability function that could dependent on y_1. Then,*

$$\mathcal{I}^{\text{VINCE}}(X;Y_1) = \mathbb{E}_{p(x,y_1)q(y_{2:K}|y_1)}\left[\left(\prod_{i=2}^{K}\frac{1}{r(y_i|y_1)}\right)\log\frac{e^{f_{\theta,\phi}(x,y_1)}}{a(x,y_{1:K})}\right] - K + 1$$

Proof. Note $\frac{p(y_i)}{q(y_i|y_1)} = \frac{1}{r(y_i|y_1)}$. □

Example 1. *Given $y_1 \sim p(y_1)$, a similarity function $f : Y \times Y \to [0,1]$, and a threshold $t \in (0,1]$, accept any sample $y_i \sim p(y_i)$ if $f(y_1, y_i) \leq t$.*

Example 2. *Given $y_1 \sim p(y_1)$, a similarity function $f : Y \times Y \to [0,1]$, and a lower threshold $l \in [0,1]$ and an upper threshold $u \in [l,1]$, accept a sample $y_i \sim p(y_i)$ if $l \leq f(y_1, y_i) \leq u$.*

We now introduce the second family.

Corollary 3.1.2. *Fix $y_1 \sim p(y_1)$. Assume a variational family for q that we can score and sample from. Assume that $c_i = \inf_{y_i}\{\frac{p(y_i)}{q(y_i|y_1)}\}$ exists. Let $c = \prod_{i=2}^{K} c_i$. Then, we can lower bound the VINCE estimator as*

$$\mathcal{I}^{\text{VINCE}}(X;Y_1) \geq \mathbb{E}_{p(x,y_1)q(y_{2:K}|y_1)}\left[c\log\frac{e^{f_{\theta,\phi}(x,y_1)}}{a(x,y_{1:K})}\right] - K + 1$$

Proof. Note $\prod_{i=2}^{K} \frac{p(y_i)}{q(y_i|y_1)} \geq \prod_{i=2}^{K} c_i$. $\square$

Example 3. *Let $q(y)$ be the uniform over the support of $p(y)$.*

Example 4. *Let $\tilde{q}(y) = p(y)/t$ where $t \in \mathbb{R}^+$. $q(y) = \tilde{q}(y)/\int_y q(y)$.*

A final note about Corollaries 3.1.1 and 3.1.2: if we use the VINCE estimator as an optimization objective (as is done in unsupervised learning), the constant terms are negligible, and both Corollaries can be simplified to the same function

$$\mathcal{I}^{\text{VINCE}}(X;Y_1) \propto \mathbb{E}_{p(x,y_1)}\left[f_{\theta,\phi}(x,y_1) - \mathbb{E}_{q(y_{2:K}|y_1)}\left[\log \frac{1}{K}\sum_{i=1}^{K} e^{f_{\theta,\phi}(x,y_i)}\right]\right] \tag{4}$$

e.g. ignore $r(y_i|y_1)$ or c and the $-K+1$ terms. Eq. 4 is how InfoNCE is commonly presented in machine learning literature [19, 13], although without justification — recent papers have suggested that using arbitrary variational distributions to draw negative samples maintains a lower bound on mutual information. While we have not found a proof of this, we have formally described a (restricted but still flexible) family of distributions that we know preserve its relation to mutual information. We will see in Sec. 5 that these two classes already describe several representation learning algorithms, including ones from [13].

4 Relations to Representation Learning

Bounds on mutual information have been widely used in unsupervised learning as training objectives [1, 10, 13]. While there has been plenty evidence of experimental success, a few questions come to mind.

First, *why does maximizing mutual information result in representations useful for downstream tasks?* Apriori, estimating mutual information and representation learning appear to be unrelated goals. One can easily imagine the former to result in meaningless embeddings. The connection between the two only exists because of an inductive bias introduced by the design of the witness function. Since $f_{\theta,\phi}(x,y)$ encodes x and y independently before the dot product, the resulting representations are forced to map the two inputs to the same space where mutual information can be measured as a linear operation. Thus, the encoders bear the near-full burden of estimating mutual information, which requires capturing nuanced semantic relationships about the random variables at hand. Consider a different design of the witness function: $f_{\theta,\phi,\psi}(x,y) = h_\psi(g_\theta(x), g_\phi(y))$ where h_ψ is an arbitrarily complex function. In the worst case, it is possible that g_θ and g_ϕ do very little (essentially identity functions) whereas h_ψ assumes the bulk of the burden of learning – representations learned in this setting would likely struggle in downstream tasks.

Second, representation learning has been focused on optimizing bounds on $\mathcal{I}(X;Y)$ for the special case where X and Y represent two *views* of the same data. *What exactly is a "view"?* Previous work has been very vague about its definition. In language, a view of a sentence may be a contiguous span of the sentence.

In vision, a view may be a cropped image or an image with pixel-level noise added. *What distinguishes a good view from a bad view?* In BERT, the masked language model objective uses a very particular algorithm for masking a token. For images, algorithms use ad hoc sets of image augmentations. Ideally, we would like understand what properties of a view make it good for representation learning. If we could do that, we could choose the best views for each domain and problem in an educated manner.

Third, since a view is a transformation on input data x, mutual information in this setting is between a random variable X and itself. *As $\mathcal{I}(X;X)$ is a trivial quantity, how is this a desirable objective?.* What role does a "view" play in optimization? For example, imagine a image view being a horizontal flip (a common augmentation used in these algorithms). On a high level, optimizing mutual information in this setting feels intuitively wrong. Next, we address some of these questions as well as leaving some open for future work.

4.1 Notation

We begin with some notation for us easily describe operations involving views. Let v_x be a set of 'views of the data point x. Each datum x will have its own set v_x. All elements of v_x have the same type and shape as x. In particular, we assume $x \in v_x$. In the language example, v_x might specify all copies of the sentence x with different tokens masked. In vision, v_x might specify all crops of the image x. Without loss of generality, assume the size of the set $|v_x|$ are the same for all x, denoted $|v|$. Define a sequence of indices $\mathcal{A}$ that assigns an arbitrary ordering to a set v_x. Note that the ordering is the same for all such v_x. Define an indexing function v that maps x and an index $a \in \mathcal{A}$ to the a-th element of v_x. In other words, $v(x,a) = v_x[a]$. A final notational point: when we write $v(x,a)$ versus $v(x,b)$, we mean to choose two different views. We will write $v(x,a)$ twice when we wish to refer to the same view twice. We write the VINCE bound, $\mathcal{I}^{\text{VINCE}}(X;X)$, for views as

$$\mathbb{E}_{p(x_1)}\left[f_{\theta,\phi}(v(x_1,a),v(x_1,b)) - \mathbb{E}_{q(x_{2:K}|x_1)}\left[\log \frac{1}{K}\sum_{i=1}^{K} e^{f_{\theta,\phi}(v(x_1,a),v(x_i,c))}\right]\right] \tag{5}$$

where $a,b,c \in \mathcal{A}$. We sample views uniformly from $\mathcal{A}$. In Eq. 5, $p(x_1)$ is assumed to be a true underlying data distribution; if we are given a training dataset, each example is therefore an i.i.d. sample from $p(x_1)$. In the following subsection, we make three observations about the way views are used in practice.

4.2 Desirable Properties of a View

We list three necessary conditions for a useful view. In our toy and real experiments, we will verify each of these properties through lesioning.

Views as Information Bottleneck We can frame views as a device for information bottleneck that force the encoders to estimate mutual information under missing information: the stronger the bottleneck, the more robust the representation. If we choose the views v_x, such that the amount of information in each view of x is variable, then the mutual information $\mathcal{I}(X;X)$ is not obvious. Merely bottlenecking information is not sufficient; different views should differ in the information they contain. In this scenario, the encoders will learn representations that are "invariant" to the many views in v_x. Consider the training paradigm for optimizing Eq. 5: every epoch, for a given x, we must choose two views randomly: a, b. Over infinite epochs, this paradigm *amortizes*[2] the parameters of the encoders (ϕ, θ) over all possible pairs of views in v_x. This compounds the amount of information that must be compressed in the encoded representation. Although this introduces an amortization gap since ϕ and θ cannot overfit to two particular views, it encourages a more abstract representation invariant to several transformations on the data, which we may prefer to have for downstream tasks. For instance, different croppings of an image can enable attention to the objects present in the scene.

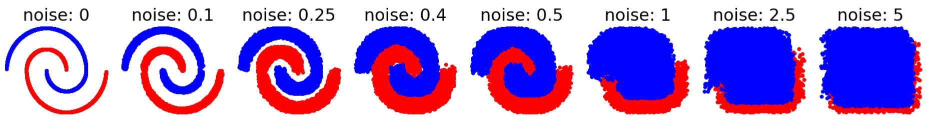
Identifiability of Views There is, of course, a limit to how much information we can bottleneck. Given an image, imagine we scramble all pixels to random values. This would clearly make it difficult for the witness function to properly access similarity but so much so that no meaningful representation can be learned. To avoid this, we generally desire views of a datum to be *identifiable* to that datum alone. In other words, the elements of view set v_x for a data point x should not appear in the view set v_y for any y. Given a element from $v(x, a) \in v_x$, it should be deterministic to find which x is associated with $v(x, a)$. Consider MLM or popular image augmentations: a text corpus is rich enough such that masking tokens in a sentence does not make it ambiguous which sentence was masked; likewise, cropping an image or adding color jitter does not obscur the identity of the original image.

A Domain Restriction on Views A more subtle but important property of views: for a datum x, a view $v(x, a)$ (for any $a \in \mathcal{A}$) must be in the same domain as x. For instance, the view of an image cannot be a piece of text, it must also be an image. In this sense, we can consider x to be a *priviledged view* (of itself). Because amortizing over views increases the generalization capability of our encoders to elements of v_x (and since $x \in v_x$), we expect amortization to result in a good representation for the priviledged view x. The *same-domain assumption* is worth highlighting: if we were to choose views of a different dimensionality or modality, we would not be able to encode (untransformed) images from a test set into our representation. If the point of optimizing mutual information is to build a representation, this would defeat the purpose.

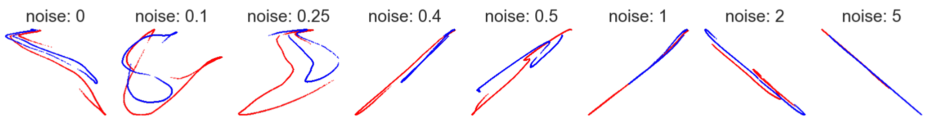
[2]In reality, the encoders are doubly-amortized over all possible values of X as well.

4.3 A Toy Example

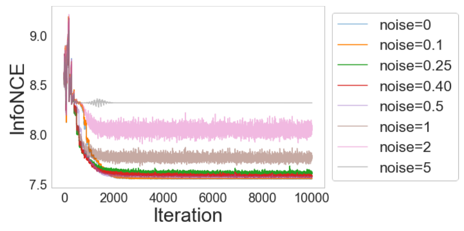
Our hypothesis thus far, is that the choice of views is extremely important. A poor choice will render this approach of representation learning obsolete. To show this, we consider a toy example where the data and the learned representation are in two dimensions. Later, we conduct an analogous experiment on ImageNet with Instance Discrimination [24].



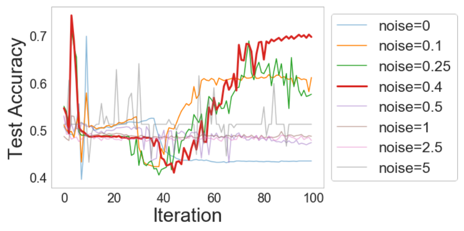
(a) Sampled views of training examples with varying levels of noise



(b) Learned 2D representations of training examples with varying levels of noise



(c) InfoNCE loss over 10k iterations



(d) Test accuracy on transfer task

Figure 1: *Double spiral synthetic example.* We apply the IR algorithm to a dataset of points along two intertwined spirals. No labels as to which spiral a point belongs to are given during training. We define a “view" to be Uniform noise added to a example (x, y). (a) We vary the amount of noise to measure the impact of the views we choose. (b) We choose a 2D representation so that we can visualize the embeddings. (c) shows the training losses over time — unsurprisingly, with more noise, the mutual information is lower. (d) To measure the quality of a representation, we use it on a transfer task to predict which spiral a point belongs to (we restrict this predictive model to be linear). In order to do well on this task, the learned representation has to “linearize" the spirals.

Consider two interweaved spirals in two dimensions, one spinning clockwise and one spinning counter-clockwise (see Fig. 1a). A data point $x = (c_x, c_y)$ is a 2D coordinate. Define the view set v_x for x as Uniform noise added to each coordinate $(c_x + \varepsilon_x, c_y + \varepsilon_y) \in v_x$ where $\varepsilon_x, \varepsilon_y \sim U(0, \eta)$. We vary η from 0, where we only have the priviledged view (i.e. $v_x = \{x\}$) to 5, where every point is essentially mapped to every point possible in the domain. The encoders are a 5-layer MLP with 128 hidden dimensions and map inputs x to a 2D representation. We use 4096 negative samples drawn from the training dataset to optimize InfoNCE with SGD using momentum 0.9, weight decay 1e-5, batch size 128, and learning rate 0.03 for 10k iterations (100 epochs through training data). A view in this example, satisfies both the information bottleneck (assuming $\eta > 0$) and same-domain constraints.

We hope to learn a representation that can capture spatial information. To measure the quality of a representation, we measure the performance on the transfer task of predicting which spiral each point belongs to. Critically, we restrict the predictive model for the transfer task to be linear (we use logistic regression with the representations as the sole input). The information required to disentangle the two spirals must be linearly separable in the representations. No labels regarding spiral identity are given during training.

Fig. 1 plots the learned representations (b), the training losses (c), and the transfer accuracy on a test set (d). With no views ($\eta = 0$), the training loss is minimized fairly well but the transfer accuracy is quite poor, near chance. As the level of noise increases approaching 0.4, we find steady improvements to transfer accuracy, reaching 0.70 on a held-out test set. As the noise level surpasses 0.4, we see that transfer accuracy recedes back to chance (0.5), and that the training loss converges to a higher number (a lower estimate of mutual information). Looking at the representations, only the best performing model ($\eta = 0.4$) has a (mostly) linear separation between the red and blue lines (representations of the two spirals). Although other noise levels have linear representations ($\eta = 1, 2, 5$), they are unable to distinguish between the two classes. Lower levels of noise led to representations that still largely resemble nonlinear spiral-like functions, which is difficult for logistic regression to divide.

We want to highlight a few additional takeaways from this toy setting: (1) Learning a good representation and truly estimating mutual information are two different problems. We do not care how much the training loss actually decreases since we are not interesting in the true underlying mutual information. We do not choose an estimator based on its tightness. (2) While we want an information bottleneck, we cannot pick views that hide too much information. It is easy to see this in the extreme: if we allow a point to jump to any other coordinate in the domain, we obscure any signal and the encoders cannot possibly compensate. A good heuristic for choosing views is identifiability. A noise level of 0.4 is on the border of which any greater loses identifiability between the two spirals.

5 A Look at Existing Algorithms

We show that many representation learning algorithms can be cast as maximizing mutual information between views: Masked Language Models, Deep InfoMax, Instance Discrimination, and Local Aggregation. In particular, we pay attention to how each algorithm defines their view sets and negative samples.

5.1 Masked Language Models

We start with the masked language modelling (MLM) objective [3] but note that connections to mutual information have been made for many language models [13, 17]. We merely suggest a few edits below.

Given a sentence of T tokens, $x = \{w_1, \ldots, w, \ldots, w_T\}$, the MLM objective randomly chooses 15% of the tokens to replace with either a mask token, a random token, or leave it unchanged. Call the latter token $\hat{w}$. For each $\hat{w}$, the objective consists of predicting the original token from the masked sentence, $\mathbb{E}_{p(x)}[p(w|\hat{x})]$ where $\hat{x} = \{w_1, \ldots, \hat{w}, \ldots, w_T\}$. Kong et. al. [13] showed that the MLM objective can be rewritten as (in our terminology):

$$\mathcal{I}^{\text{MLM}}(X; X) = \mathbb{E}_{p(x_1)}\left[f_{\theta,\phi}(v(x_1, a), u(x_1, a)) - \log \frac{1}{|\mathcal{V}|} \sum_{w \in \mathcal{V}} e^{f_{\theta,\phi}(v(x,a),w)}\right]$$

where for a datum x, the view set v_x contains replicas of sentence x with a different token masked out. Note that by construction, $x \in v_x$ (i.e. when the token to be masked is left unchanged). Special to MLM, we introduce a separate view set u_x implicitly defined by a new indexing function u. This view set contains only the masked out tokens. Notice that the witness function takes in an element from v_x and an element from u_x using the same index a — a masked sentence uniquely specifies a masked out token. In other words, there is a bijection between v_x and u_x. Only in negative sampling do we choose a different masked out token, represented by w. In detail, the witness function can be written as $f_{\theta,\phi}(v(x, a), u(x, a)) = g_\theta(v(x, a))^T g_\phi(u(x, a))$. The first encoder g_θ extracts a hidden vector for a masked sentence using a Transformer. The second encoder g_ϕ is an embedding mapping each token in the vocabulary to a vector (e.g. one hot embedding).

Given a masked sentence and its true masked out token, the MLM algorithm compares the similarity of their representations using “negative samples" by enumerating over all possible masked out tokens in the training vocabulary, denoted $\mathcal{V}$. Kong et. al. [13] noted that doing so is quite similar to the InfoNCE estimator. However, because the InfoNCE is only a proper lower bound when sampling from $\prod_{i=2}^{|\mathcal{V}|} p(x_i)$, and since different tokens have different marginal probabilities, using all the words in the vocabulary with equal probability cannot directly be framed as lower bounding mutual information. To be faithful to InfoNCE, we should with higher probability choose the word “the" as our negative sample than most other words.

The benefit of the VINCE estimator is the freedom to choose from a larger family of distributions. We can view enumerating over the vocabulary as $|\mathcal{V}|$ independent samples from the Uniform distribution over all possible tokens in the training corpus. Equivalently, assuming all tokens in a sentence x are equally likely to be the masked out token, we can choose $q(x_{2:|\mathcal{V}|}) = \prod_{i=2}^{|\mathcal{V}|} q(x_i)$ where the distribution $q(x)$ is proportional to the inverse empirical frequency of the tokens in sentence x. In other words, we choose q to make all tokens in the vocabulary equally probable of being sampled. By VINCE, we know this choice bounds mutual information (See Ex. 3). In other words,

$$\mathcal{I}^{\text{VINCE}} = \mathbb{E}_{p(x_1)}\left[f(v(x_1,a),u(x_1,a)) - \mathbb{E}_{q(x_{2:|\mathcal{V}|})}\left[\log\frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|} e^{f(v(x_1,a),u(x_i,b)}\right]\right]$$
$$= \mathcal{I}^{\text{MLM}}$$

5.2 Deep InfoMax

Of the many visual representation learning algorithms [5, 4, 10, 24, 26], only Deep InfoMax [10] is explicitly tied to mutual information. Deep InfoMax (DIM) extends the infomax principle [14, 2], which learns representations by maximizing a lower bound on the mutual information between the inputs and outputs (or activations) of a neural network.

Let X be a random variable representing an image and $Y = E_\psi(X)$ be the random variable representing the output of a neural network E with realizations of X as input. The DIM objective can be written as:

$$\mathcal{I}^{\text{DIM}}(X;X) = \mathbb{E}_{p(x_1)}\left[f_{\theta,\phi}(x_1,E_\psi(x_1)) - \mathbb{E}_{p(x_{2:K})}\left[\log\frac{1}{K}\sum_{i=1}^{K} e^{f_{\theta,\phi}(x_1,E_\psi(x_i))}\right]\right]$$

where DIM draws independent negative samples from the training dataset where each example is given equal probability of being chosen. If we believe the training examples were sampled i.i.d. from the true marginal, then we are effectively drawing negatives from $p(x_{2:K})$. InfoNCE ensures that this choice of distribution is a valid lower bound on mutual information. The witness is defined as $f_{\theta,\phi}(x,E_\psi(x)) = g_\theta(x)^T g_\phi(E_\psi(x))$ where neural network g_θ ingests an image and g_ϕ ingests an activation.

However, $\mathcal{I}^{\text{DIM}}(X;X)$ is in some ways undesirable: Since X is the raw image, the DIM objective might encourage trivial representations that noisily memorize "useless" pixel-level information whereas an ideal representation would capture higher-level semantic meaning. To address this, Hjelm et. al. [10] also introduces *Local Deep InfoMax* (LDIM), which maximizes the mutual information between a "global" feature vector and many "local" feature vectors, avoiding the raw image entirely. Define a "local" neural network $L_{\psi_1} : X \to \mathbb{R}^{N\times M\times M}$ that maps an image into N different spatial feature maps of size M by M. Next, define a "global" neural network $G_{\psi_2} : \mathbb{R}^{N\times M\times M} \to \mathbb{R}^d$ that maps the set of spatial feature filters to a global feature vector. For shorthand, let $E_\psi(x) = (G_{\psi_2} \circ L_{\psi_1})(x)$ where $\psi = \{\psi_1, \psi_2\}$. The LDIM bound maximizes

$$\mathcal{I}^{\text{LDIM}}(X;X) = \frac{1}{M^2}\sum_{m=1}^{M^2}\mathcal{I}^{\text{NCE}}(L_{\psi_1}^{(m)}(X);E_\psi(X))$$
$$\mathcal{I}^{\text{NCE}}(\cdot;\cdot) = \mathbb{E}_{p(x_1)}\left[L_{\psi_1}^{(m)}(v(x_1,a))^T E_\psi(v(x_1,a)) - \mathbb{E}_{p(x_{2:K})}\left[\log Z(x_{1:K},m)\right]\right]$$

where the partition $Z(x_{1:K},m) = \sum_{i=1}^{K}\exp\{L_{\psi_1}^{(m)}(v(x_1,a))^T E_\psi(v(x_i,b))\}$ and $L_{\psi_1}^{(m)} : X \to \mathbb{R}^{M\times M}$ represents indexing the m-th spatial filter. We slightly abuse notation with $L^{(m)}_{\psi_1}(X)$ and $E_\psi(X)$ to define random variables whose values are put through the deterministic $L^{(m)}_{\psi_1}$ and E_ψ functions, respectively.

As DIM and LDIM make use of InfoNCE, their connections to mutual information are obvious. What is more interesting is amortization with LDIM. The view set v_x for an image x is defined by the composition of image augmentations used by [10] i.e. 64 x 64 random cropping and horizontal flip. As the authors found, data augmentation is *critical* to learning a good representation and good transfer task performance. However, no explanations were given for why data augmentation is so important. We can interpret this phenomenon via amortization: recall that because choosing a view is random, every epoch will generate a different view on an image x (e.g. different crops). This forces the encoder representations to generalize to wider range of images, preventing it from overfitting to a single privileged view. We can additionally consider the different feature maps as different "views" of $L^{(m)}_{\psi_1}(X)$, suggesting that even when the authors in [10] did not explicitly augment the images, the LDIM algorithm implicitly amortizes over M^2 views.

5.3 Instance Discrimination and Local Aggregation

The mutual information perspective of representation learning offers a practical lens in interpreting what various algorithms do. In particular, we can compare methods like MLM and DIM based on what they choose for their witness function, proposal distributions, and view sets. In this section, we extend the family of algorithms encompassed by this mutual information perspective to two new representation learning algorithms: Instance Discrimination (IR) [24] and Local Aggregation (LA) [26]. We find that using mutual information, we gain many insights into (1) the differences between IR and LA, (2) why one works better than the other, and (3) a family of related vision algorithms. Below, we begin by presenting the algorithms as described by their authors. Afterwards, we present an alternative description using mutual information.

Background Instance Discrimination (IR) is the first of two vision algorithms that learn representations by classification ("discrimination") of individual examples from a set. In particular, IR generalizes the supervision training paradigm, treating each example ("instance") in the training dataset as its own class [24]. Let x_i be an image and $E_\psi(v(x_i, a))$ be an encoding where i is the index in the training dataset and $v(x, a)$ is returns the a-th index in the view set v_x. The function E_ψ is a neural network that maps the view of an image to an activation vector of dimension d. IR uses the composition of several image augmentations (random cropping, random grayscaling, random color jittering, and random horizontal flipping) to define a very large view set for every image

x. IR maximizes the following,

$$\mathcal{L}^{\text{IR}}(x_i, M) = \log p(i|x_i, M), \qquad p(i|x, M) = \frac{\exp\{E_\psi(v(x,a))^T M[i]/\tau\}}{\sum_{j=1}^{N} \exp\{E_\psi(v(x,a))^T M[j]/\tau\}} \tag{6}$$

where N is size of the training dataset, and τ is a temperature parameter to smooth the softmax, which otherwise may have issues with gradient saturation due to large number of classes. The denominator in $p(i|x, M)$ requires computing a forward pass with E_ψ for N data points which can be prohibitively expensive for a large dataset like ImageNet [21]. Instead, the authors use of a *memory bank* M to store the representations for every image (instead of recomputing each time). The representations for each image i are updated using a linear combination of the current representation and the new representation $E_\psi(v(x_i, a))$ every epoch to prevent them from growing stale.

$$M[i] = \alpha * M[i] + (1-\alpha) * E_\psi(v(x_i, a)) \tag{7}$$

where $a \in \mathcal{A}$ (the view index) changes every update, and α is hyperparameter controlling the speed of update. The notation $M[i]$ retrieves the representation for the i-th element from the memory bank. If $\alpha = 0$, the memory bank stores only the representation of the view of image x_i from the last epoch. Over epochs, the contribution of any single view decays exponentially: $\alpha, \alpha^2, \alpha^3, \ldots$. The authors find that the unsupervised representations learned with IR perform well in downstream ImageNet classification (with no prior knowledge of labels).

The second algorithm, Local Aggregation (LA), was introduced as an improvement to IR, shown in [26] to learn representations that outperform those of IR in downstream ImageNet classification. If we think of IR as learning a representation that "pushes" the representations of all other images away (i.e. discrimination), then LA biases a representation that pulls "nearby" images closer while "pushing" other images away. The LA objective is

$$\mathcal{L}^{\text{LA}}(x_i, M) = \log \frac{p(C_i \cap B_i|x_i, M)}{p(B_i|x_i, M)}, \qquad p(I|x, M) = \sum_{i \in I} p(i|x, M) \tag{8}$$

where I is an arbitrary set of indices (spanning the training dataset) containing i. The terms $p(i|x, M)$ and x_i are as defined in Eq. 6. LA includes the same image augmentations as IR as well as the memory bank M. However, LA introduces two new sets, a *background neighbor set* B_i, and a *close neighbor set* C_i. Given an image x_i, the former is defined to be the k closest examples to $M[i]$ (the current representation for image x_i stored in the memory bank) in M, measured by cosine similarity in $\mathbb{R}^d$. The latter is defined by one or more separate clustering algorithms (e.g. kNN) where the elements of C_i belong to the same cluster as $M[i]$. Intuitively, the elements of C_i should be "closer" to x_i than the elements of B_i, which acts as a baseline level of "closeness".

From the descriptions above, it is not clear how to compare these algorithms to each other and to other unsupervised learning algorithms like MLM. Several questions come to mind: *Why does LA outperform IR? What is the memory bank actually doing? What is the impact of using B_i and C_i?* One of our goals is to provide a more rigorous analysis of LA and IR. We will utilize the tools we have built with mutual information. By doing so, we hope to shed light on why these recent unsupervised visual algorithms have been so successful.

Relation to Mutual Information We aim to show that IR and LA are lower bounds on the mutual information between two *weighted sets of views* of an image. Recall that a view set v_x is defined by the image augmentations used on image x and the indexing function v maps x and an index $a \in \mathcal{A}$ to the a-th view in v_x. Let E_θ be a neural network that maps a view to vector in $\mathbb{R}^d$. Lastly, given the i-th image x_i and the memory bank M with update α, we can write

$$M[i] = \sum_{j=1}^{|v|} \alpha^{j-1} E_\theta(v(x_i, j)) \approx \sum_{j=1}^{n_\alpha} \alpha^{j-1} E_\theta(v(x_i, a_j)) \tag{9}$$

where $a_j \in \mathcal{A}$ represents the j-th randomly chosen index. Let $\mathcal{A}_{x_i} = \{a_j\}_{j=1}^{n_\alpha}$ be the set of observed indexes that contribute to memory bank entry i over n_α epochs of training. Eq. 9 poses that the first sum, which enumerates over all possible views of x_i, can then be tightly approximated by a finite sum where the size n_α is a function of the update rate α since the contribution of any individual view exponentially decays with respect to α. For instance, if $\alpha = 0$, then $n_\alpha = 1$.

Define a set $S_{x_i} = \{v(x_i, a)\}$ to contain a single random view of x_i. Define a second set $T_{x_i} = \{v(x_i, a) | a \in \mathcal{A}_i\}$ that contains the n_α randomly chosen views of x_i stored in the memory bank entry $M[i]$. Introduce indexing functions $v_S(x_i, a)$, which maps x_i to S_{x_i}, and $v_T(x_i, \mathcal{A}_i)$, which maps x_i to T_{x_i}. Finally, fix two sets of weights, $W_S = \{1\}$ and $W_T = \{1, \alpha, \alpha^2, \ldots, \alpha^{n_\alpha - 1}\}$. Then, the encoder g_θ is as follows:

$$g_\theta(v(x, \cdot), W) = \frac{1}{|S|} \sum_{k=1}^{|S|} W[k] \cdot E_\theta(S[k]) \bigg|_{S = v(x, \cdot)} \tag{10}$$

where the set $S = v(x, \cdot)$ and $S[k], W[k]$ retrieve the k-th elements of each set. Note that $g_\theta(v_S(x_i, a), W_S) = E_\theta(v(x_i, a))$. The witness function is then

$$f_\theta(v_S(x_i, a), v_T(x_j, \mathcal{A}_j)) = g_\theta(v_S(x_i, a), W_S)^T g_\theta(v_T(x_j, \mathcal{A}_j), W_T)/\tau \tag{11}$$

where τ is the temperature in Eq. 6. In other words, it is the dot product between weighted encodings of views. We can now explicitly connect IR to InfoNCE.

Theorem 5.1. *Let X be a random variable.* $\mathcal{I}^{\text{NCE}}(X; X) \propto \mathbb{E}_{p(x)}\left[\mathcal{L}^{\text{IR}}(x)\right]$.

Proof. Recall that x_i is the i-th data point in the training set and M is a memory bank. Then, we can write the IR objective in the form of InfoNCE.

$$
\begin{aligned}
\mathbb{E}_{p(x_i)}[\mathcal{L}^{\text{IR}}(x_i)] = \mathbb{E}_{p(x_i)}[\log p(i|x_i, M)] &\propto \mathbb{E}_{p(x_i)}\left[\log \frac{e^{(E_\theta(v(x,a))^T M[i]/\tau)}}{\frac{1}{N}\sum_{j=1}^N e^{(E_\theta(v(x,a))^T M[j]/\tau)}}\right] \\
&= \mathbb{E}_{p(x_i)}\left[E_\theta(v(x,a))^T M[i]/\tau - \log \sum_{j=1}^N e^{(E_\theta(v(x,a))^T M[j]/\tau)}\right] \\
&= \mathbb{E}_{p(x_i)}\left[f_\theta(v_S(x_i,a), v_T(x_i,\mathcal{A}_i)) - \log \frac{1}{N}\sum_{j=1}^N e^{(f_\theta(v_S(x_i,a), v_T(x_j,\mathcal{A}_j)))}\right] \\
&= \mathbb{E}_{p(x_i)}\left[f_\theta(v_S(x_i,a), v_T(x_i,\mathcal{A}_i)) - \mathbb{E}_{p(x_{j\neq i})}\left[\log \frac{1}{N}\sum_{j=1}^N e^{(f_\theta(v_S(x_i,a), v_T(x_j,\mathcal{A}_j)))}\right]\right] \\
&= \mathcal{I}^{\text{NCE}}(X;X) \leq \mathcal{I}(X;X)
\end{aligned}
$$

where $p(x_{j\neq i}) = \prod_{j=1,j\neq i}^N p(x_j)$. The proportionality in the first line comes from adding a $\log N$ constant. In the second to last line, we use the fact that IR chooses negative samples from the training dataset, which is equivalent to i.i.d. sampling from the true marginal $p(x)$. $\square$

Interestingly, our formulation of IR (second to last line of the proof) does not mention the memory bank. In fact, we can view the memory as an implementation detail used only for computational efficiency. There is no reliance on it from a theoretical perspective. Since re-encoding elements of $v_T(x_i, \mathcal{A}_i)$ would be expensive, the memory bank is a clever data structure that can store the running average e.g. $g_\theta(v_T(x_i, \mathcal{A}_i), W_T)$ for all i. It is not used for $g_\theta(v_S(x_i, a), W_S)$ since this set only contains one item (no need for storage).

As you may have observed, the memory bank storing a weighted sum of views makes the mutual information formulation quite involved. So, we consider the special (and simple) case of Thm. 5.1 when $\alpha = 0$. By Eq. 9, the i-th entry $M[i]$ stores only the encoding of a single view: the one randomly chosen in the last epoch. In this case, we no longer require the set notation, weights, nor the memory bank itself. We can simplify to

$$
\mathbb{E}_{p(x_i)}\left[f_\theta(v(x_i,a), v(x_i,b)) - \mathbb{E}_{p(x_{j\neq i})}\left[\log \frac{1}{N}\sum_{j=1}^N \exp\{f_\theta(v(x_i,a), v(x_j,c))\}\right]\right]
$$

where $g_\theta(v(x_i, a)) = E_\psi(v(x_i, a))$. When $\alpha = 0$, IR estimates the mutual information between two random views of x. We will show in our experiments that this version is not only simpler but performs as well on downstream tasks, questioning the usage of a memory bank.

Interpreting IR as a mutual information estimator suggests a few variations of the vision algorithm by changing the proposal distribution $p(x_{j\neq i})$. So far, we have limited ourselves to drawing from the true marginal (InfoNCE).

However, we have already shown that choosing q from a restricted family of distributions preserve a lower bound. Ideally, we would not need to see all $N-1$ negative examples; instead, if we are forced to choose a smaller set of negative examples, *choosing ones that are more difficult to discriminate* can lead to better representations. For example, with images, encoders that can distinguish between two dogs (rather than a dog and a cat) forces the learned representation to be more semantically meaningful. With this intuition, we introduce two new representation learning algorithms closely related to IR.

Definition 5.2. *Ball Discrimination.* $\mathcal{L}^{\text{BALL}}(x_i, M) = \log \frac{p(i|x_i,M)}{p(B_i|x_i,M)}$.

Definition 5.3. *Annulus Discrimination.* $\mathcal{L}^{\text{ANN}}(x_i, M) = \log \frac{p(i|x_i,M)}{p(B_i \backslash C_i|x_i,M)}$.

Using VINCE, we show that BALL and ANN both lower bound $\mathcal{I}(X;X)$.

Corollary 5.3.1. *Ball and Annulus Discrimination are both proportional to VINCE:* $\mathcal{I}^{\text{VINCE}}(X;X) \propto \mathbb{E}_{p(x)}[\mathcal{L}^{\text{BALL}}(x)]$ *and* $\mathcal{I}^{\text{VINCE}}(X;X) \propto \mathbb{E}_{p(x)}[\mathcal{L}^{\text{ANN}}(x)]$.

Proof. Expand the BALL estimator and cancel denominators.

$$\begin{aligned}
\mathbb{E}_{p(x_i)}[\mathcal{L}_\theta^{\text{BALL}}(x_i)] &= \mathbb{E}_{p(x_i)}\left[\log \frac{p(i|x_i,M)}{p(B_i|x_i,M)}\right] \\
&= \mathbb{E}_{p(x_i)}\left[\log\left(\frac{\frac{e^{(E_\psi(v(x_i,a))^T M[i]/\tau)}}{\sum_{j=1}^N e^{(E_\psi(v(x_i,a))^T M[j]/\tau)}}}{\sum_{k\in B_i}\left(\frac{e^{(E_\psi(v(x_i,a))^T M[k]/\tau)}}{\sum_{j=1}^N e^{(E_\psi(v(x_i,a))^T M[j]/\tau)}}\right)}\right)\right] \\
&\propto \mathbb{E}_{p(x_i)}\left[\log\left(\frac{e^{(E_\psi(v(x_i,a))^T M[i]/\tau)}}{\frac{1}{|B_i|}\sum_{k\in B_i} e^{(E_\psi(v(x_i,a))^T M[k]/\tau)}}\right)\right] \\
&= \mathbb{E}_{p(x_i)}\left[E_\psi(v(x_i,a))^T M[i]/\tau - \log\frac{1}{|B_i|}\sum_{k\in B_i}\left(e^{(E_\psi(v(x_i,a))^T M[k]/\tau)}\right)\right] \\
&= \mathbb{E}_{p(x_i)}\left[f_\theta(v_S(x_i,a), v_T(x_i,\mathcal{A}_i)) - \mathbb{E}_{q(x_{j\neq i}|x_i)}\left[\log\frac{1}{|B_i|}\sum_{j=1}^{|B_i|} e^{(f_\theta(v_S(x_i,a),v_T(x_j,\mathcal{A}_j)))}\right]\right] \\
&= \mathcal{I}^{\text{VINCE}}(X;X) \leq \mathcal{I}(X;X)
\end{aligned}$$

We skipped a few steps in the second to last line since it is almost identical to the proof of Thm. 5.1. In short, we need to define encoders on indexing function into sets v_S and v_T. The only difference in this setting is that negative samples come from a background set B_i rather than the entire training dataset. Thus, we cannot equate the BALL estimator to InfoNCE. Instead we must rely on VINCE, which allows us to encapsulate two interesting properties of BALL.

First, recall that the background set B_i contains the closest examples to $M[i]$ by cosine similarity. Thus, negative samples x_j are *conditioned* on the original sample x_i. To handle this, we need a conditional variational distribution

$q(x_{j\neq i}|x_i) = \prod_{j=1,j\neq i}^{|B_i|} q(x_j|x_i)$. Second, the negative samples are still drawn from $p(x_i)$ i.e. they are from the training dataset. However, not all examples are valid: only the ones “close" to the representation of x_i are considered. To represent this, we define $q(x_j|x_i)$ as the distribution induced by rejection sampling where we accept any sample whose cosine similarity is above a threshold (implicitly set by the user). This is exactly Ex. 1.

We omit the proof for ANN as it is nearly identical. Recall that ANN draws negative samples from B_i C_i i.e. is does not use examples too close to x_i. To model this, fix the proposal $q(x_j|x_i)$ as the rejection sampled distribution where we accept any sample whose cosine similarity is between a lower and upper threshold. This is exactly Ex. 2. As such, we can still show that BALL and ANN lower bounds mutual information. □

Finally, we move to Local Aggregation, which we will show is closely tied to BALL. We start with a simplified version where we assume that the closest neighbor set contains only the index of the current image, $C_i = \{i\}$. We call this estimator LA-SIMPLE. We visit the general version after.

Theorem 5.4. *Let X be a random variable. Let the close neighbor set $C_i = \{i\}$. Then $\mathcal{I}(X;X) \geq \mathcal{I}^{\text{LA-SIMPLE}}(X;X)$.*

Proof. We reduce LA-SIMPLE to BALL.

$$\mathcal{L}^{\text{LA-SIMPLE}}(x_i, M) = \log \frac{p(C_i \cap B_i|x_i, M)}{p(B_i|x_i, M)} = \log \frac{p(i|x_i, M)}{p(B_i|x_i, M)} = \mathcal{L}^{\text{BALL}}(x_i, M)$$

□

Corollary 5.4.1. *For any close neighbor set C_i containing i, $\mathcal{I}(X;X) \geq \mathbb{E}_{p(x)}[\mathcal{L}^{\text{LA}}(x, M)]$.*

Proof. Given two random variables X and Y, let Y^ε be Y with some random noise added. By definition, $\mathcal{I}(X;Y) \geq \mathcal{I}(X;Y^\varepsilon)$ since any noise should decrease information between the variables. We will apply this intuition to show $\mathbb{E}_{p(x)}[\mathcal{L}^{\text{LA-SIMPLE}}(x, M)] \geq \mathbb{E}_{p(x)}[\mathcal{L}^{\text{LA}}(x, M)]$.

By design, the images in the set $\{x_k | k \in C_i\}$ should be the most semantically close to x_i out of the full training dataset. Thus, we can view each x_k as x_i with some noise added (this may be pixel level noise or semantic noise). For example, if x_i is an image of a dog, x_k may be the same dog from a different point of view, or a different dog with similar visual features. Formally, let $x_i^\varepsilon = x_i + \varepsilon$ be the i-th image with noise added. Fix i and assume that for all $k \in C_i$, there exists an ε such that some $x_k \approx x_i^\varepsilon$. Then,

$$\begin{aligned}\mathbb{E}_{p(x_i^\epsilon)}\left[E_\theta(v(x_i^\epsilon, a))^T M[i]/\tau\right] &\approx \frac{1}{N}\sum_{j=1}^{N} E_\theta(v(x_i^{\varepsilon_j}, a))^T M[i]/\tau \\ &\approx \frac{1}{|C_i \cap B_i|}\sum_{k \in C_i \cap B_i} E_\theta(v(x_k, a))^T M[i]/\tau\end{aligned}$$

where we pick the number of samples $N = |C_i \cap B_i|$. In other words, we can view the elements of the close neighbor set as a Monte Carlo approximation of an expectation with respect to the marginal distribution over X^ε, denoted $p(x^\varepsilon)$. With this formulation, we can complete the proof:

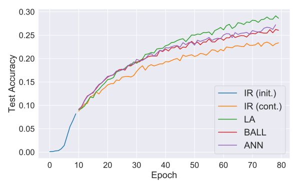
$$\begin{aligned}
\mathcal{L}^{\text{LA}}(x_i, M) &= \log \frac{p(C_i \cap B_i | x_i, M)}{p(B_i|x_i, M)} \propto \log \frac{\frac{1}{|C_i \cap B_i|}\sum_{k \in C_i \cap B_i} p(k|x_i, M)}{p(B_i|x_i, M)} \\
&= \log \frac{\frac{1}{|C_i \cap B_i|}\sum_{k \in C_i \cap B_i} e^{(E_\theta(v(x_k, a))^T M[i]/\tau)}/Z}{p(B_i|x_i, M)} \\
&\approx \log \frac{\mathbb{E}_{p(x_i^\epsilon)}[e^{(E_\theta(v(x_i^\epsilon, a))^T M[i]/\tau)}/Z]}{p(B_i|x_i, M)} \\
&\leq \log \frac{\mathbb{E}_{p(x_i)}[e^{(E_\theta(v(x_i, a))^T M[i]/\tau)}/Z]}{p(B_i|x_i, M)} \\
&= \log \frac{p(i|x_i, M)}{p(B_i|x_i, M)} = \mathcal{L}^{\text{LA-SIMPLE}}(x_i)
\end{aligned}$$

where Z is the normalization constant. The inequality in line 3 holds since adding noise to a vector should decrease correlation with any other vector. $\square$

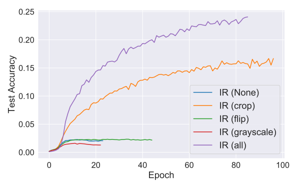
We can now say something concrete about the differences between IR and LA. Interestingly, it appears that LA is a looser bound on mutual information than BALL (or IR) which we might expect to be a disadvantage. But as we have stated, tightness on the bound and learning a good representation are two different objectives. In return for looseness, LA expands the view set for each image with nearby images in the training dataset, providing more of an information bottleneck useful for classification. Second, LA draws negative samples from a smaller set than IR. The pool that LA uses contains images more similar to x_i, making the discrimination objective more difficult. From our analysis in Sec. 4, we know that performance of an algorithm in this family is determined by its views and its negative samples. If we cast elements of C_i as x_i with the proper levels of semantic noise, then like our toy experiments, these additional views can greatly improve performance. Along the same lines, forcing BALL, ANN, and LA to distinguish between semantically closer negative samples specifies the information that the encoders should attend to. Like DIM, using unrelated images for negative samples runs the risk of encoders capturing low level image features that are less useful for representation learning.

ImageNet Experiments To test these intuitions, we explore IR, BALL, ANN, and LA pre-trained on ImageNet with a transfer task of ImageNet classification. We conducted several experiments lesioning different components of the methods to test the mutual information perspective. In particular, how do different view sets, proposal distributions for negative samples, and hyperparameters affect learning? We ran three main sets of experiments. First, we compare the transfer task performance of representations learned with IR, BALL, ANN, and

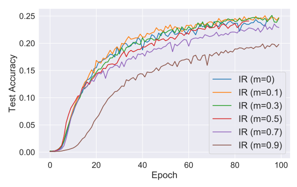
LA. Second, we compare the performance of IR when we limit the view set i.e. augmentations. Third, we compare the performance of IR with different update parameters α. Fig. 2 show the results. Below, we review the general patterns they uncover. See Sec. D for experiment details.



(a)



(b)



(c)

Figure 2: *Experimental Results* – (a) Comparison of different visual representation learning algorithms using ImageNet classification error. A large portion of the improvement of LA over IR is explanable by the proposal distribution shown by BALL and ANN. (b) Lesion experiments to quantify the contributions of different augmentations on representation quality. Critically, many of the augmentations (e.g. flip) are trivial and do not mask any of the information. The more difficult augmentations, such as cropping, have much more of an effect on the learned representation. Training runs in (b) have a patience of 10 epochs. (c) Effect of the memory bank update parameter m on transfer accuracy: as m approaches 0, the memory bank stores only the view of each image seen in the last epoch. Thus, the IR algorithm effectively maximizes the mutual information between the current view and the stored view.

- **The proposal distribution and view set are equally important.** The differences between LA and IR amount to (1) the proposal distribution and (2) the view set. We would like to disentangle which of the two, if not both, are responsible for the improved performance. Fig. 2a shows that BALL and ANN, which only add (1) but not (2), already show improvement over IR, with ANN performing slightly better than BALL. However, LA outperforms ANN by about the same amount that ANN outperformed IR. Thus, how we pick negative samples and views seem to have disjoint but equally important contributions to a good representation.

- **The choice of augmentations matters greatly.** IR [24] and LA [26] both use a standard set of image augmentations without much emphasis on their importance. However, as shown in Fig. 2b, the choice of augmentation is extremely impactful on the quality of the representation, measured by classification accuracy on ImageNet. If we used no augmentations at all, the model plateaus at 2% accuracy. Further, we find compelling evidence that not all augmentations are equal. Some like flip, do not improve learning (like we anticipated). Others like random crop “mask information" and thus, lead to useful representations and better classification performance.

As we have already emphasized the importance of views in representation learning, this should no longer come as a surprise.

- **We can simplify the Instance Discrimination algorithm** The use of the memory bank and several technical tricks like temperature to make the non-parameteric softmax stable has been credited to the success of IR. However, if we reduce IR to InfoNCE, we find that many of these complexities can be greatly simplified. We no longer need to explicitly compute the softmax as InfoNCE and VINCE only require a LOGSUMEXP. We have observed that τ plays a negligible role: whether we set it to 0.07 as previous papers do or 1, the representations learned with InfoNCE still perform as well on ImageNet classification. By varying α from 0 to near 1, we find that storing fewer representations in the memory bank performs better. As m approaches 1, we are effectively storing all images from our training dataset in the memory bank in which case the weighted combination is going to look very similar to random noise. We saw how this negatively impacted performance in the toy example. With $m = 0$, IR does just as well as the default ($m = 0.5$). This suggests that we do not require a memory bank but merely can choose two random views at every iteration. Doing so would not only simplify the theoretical interpretation, but also the practical implementation.

6 Future Directions

We have presented an interpretation of recent representation learning algorithms using mutual information between “views". This led to interesting insights that allowed us to more formally compare and contrast these various self-supervised algorithms. In particular, the mutual information angle helped explain differences in performance between LA and IR. This perspective has opened many unanswered questions. To name a few: IR and LA are no longer very different than MLM – are there possible extensions to multimodal domains that utilize both objectives? Also, since the choice of proposal distribution and view sets are both important and difficult to make, can we jointly learn them? In the supplement, we provide some thoughts on how to learn the proposal q.

References

[1] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[2] Jean-Fran cois Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal processing letters*, 1997.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[5] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017.

[6] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

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

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

[11] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

[12] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[13] Lingpeng Kong, Cyprien de Masson d'Autume, Wang Ling, Lei Yu, Zihang Dai, and Dani Yogatama. A mutual information maximization perspective of language representation learning. *arXiv preprint arXiv:1910.08350*, 2019.

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

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[16] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[18] Allen Nie, Erin D Bennett, and Noah D Goodman. Dissent: Sentence representation learning from explicit discourse relations. *arXiv preprint arXiv:1710.04334*, 2017.

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

[20] Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019.

[21] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[24] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*, 2019.

[25] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[26] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[27] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[28] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012, 2019.

A Proof of InfoNCE

Lemmas A.1, A.2 and Thm. A.3 are largely taken from [20]. We will start with a simpler estimator of mutual information, known as the Barber-Agakov estimator, and work our way to InfoNCE.

Lemma A.1. *Define the (normalized) Barber-Agakov bound as*

$$\mathcal{I}^{\mathrm{BA}}(X;Y) = \mathbb{E}_{p(x,y)}\left[\log q(x|y) - \log p(x)\right] \tag{12}$$

where $q(x|y)$ is a variational distribution approximating $p(x|y)$. Then $\mathcal{I}(X;Y) \geq \mathcal{I}^{\mathrm{BA}}(X;Y)$.

If we let $q(x|y) = \frac{p(x)e^{f(x,y)}}{Z(y)}$ *where* $f(x, y)$ *is the witness function and* $Z(y) = \mathbb{E}_{p(x)}[e^{f(x,y)}]$ *is the partition, define the unnormalized Barber-Agakov bound as*

$$\mathcal{I}^{\text{UBA}}(X;Y) = \mathbb{E}_{p(x,y)}\left[f(x,y) - \log Z(y)\right] \tag{13}$$

Then, $\mathcal{I}(X;Y) \geq \mathcal{I}^{\text{UBA}}(X;Y)$*.*

Proof. Starting with $\mathcal{I}^{\text{BA}}(X;Y)$, multiply and divide $\mathcal{I}(X;Y)$ by $q(x|y)$:

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

For the unnormalized bound, use the definition for $q(x|y)$,

$$\begin{aligned}
\mathcal{I}(X;Y) &\geq \mathbb{E}_{p(x,y)}\left[\log\frac{q(x|y)}{p(x)}\right] = \mathbb{E}_{p(x,y)}\left[\log\frac{p(x)e^{f(x,y)}}{p(x)Z(y)}\right] \\
&= \mathbb{E}_{p(x,y)}\left[f(x,y) - \log Z(y)\right] = \mathcal{I}^{\text{UBA}}(X;Y)
\end{aligned}$$

□

The log-partition $\log Z(y)$ is difficult to evaluate. The next estimator bounds it instead.

Lemma A.2. *Define the Nguyen-Wainwright-Jordan bound as*

$$\mathcal{I}^{\text{NWJ}}(X;Y) = \mathbb{E}_{p(x,y)}\left[f(x,y)\right] - \mathbb{E}_{p(y)}\left[\frac{1}{e}Z(y)\right] \tag{14}$$

where $Z(y)$ *is defined as in Lemma A.1. Then* $\mathcal{I}(X;Y) \geq \mathcal{I}^{\text{NWJ}}(X;Y)$*.*

Proof. We will be using the following upper bound $\log Z(y) \leq \frac{Z(y)}{a(y)} + \log a(y) - 1$ where the function $a(y)$ must be positive. The bound is tight if $a(y) = Z(y)$. Use this on $\mathcal{I}^{\text{UBA}}(X;Y)$:

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

Now, choose $a(y) = e$. Then,

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

□

Finally, we are ready to show that the InfoNCE estimator lower bounds mutual information. One of primary differences between the NWJ and the InfoNCE estimators is that the latter reuses the sample y from the joint $p(x, y)$ in estimating the partition.

Theorem A.3. *Let X, Y_1 be two random variables. Let $Y_2, \ldots Y_K$ be auxiliary random variables distributed as Y_1 is. Let $K-1$ be the number of negative samples. Define $p(x, y_1)$ as the joint distribution for X and Y_1. Define $p(y_{2:K}) = \prod_{j=2}^{K} p(y_j)$ as the distribution over negative samples. Fix $p(y_j) = p(y_1)$, the marginal distribution for Y_1. Recall the InfoNCE estimator*

$$\mathcal{I}^{\text{NCE}}(X; Y_1) = \mathbb{E}_{p(x,y_1)}\left[f_{\theta,\phi}(x, y_1) - \mathbb{E}_{p(y_{2:K})}\left[\log \frac{1}{K}\sum_{j=1}^{K} e^{f_{\theta,\phi}(x,y_j)}\right]\right] \tag{15}$$

. Then, $\mathcal{I}(X; Y_1) \geq \mathcal{I}^{\text{NCE}}(X, Y_1)$. That is, the InfoNCE objective is a lower bound for the mutual information between X and Y_1.

Proof. We will show that the InfoNCE bound arises from augmenting y_1 with a set of auxiliary samples $y_2, ..., y_K$ and employ the Nguyen-Wainwright-Jordan bound.

Let $Y_{2:K}$ be a set of random variables distributed independently according to $p(y_1)$. As Y_1 and $Y_{2:K}$ are independent, $\mathcal{I}(X; Y_1) = \mathcal{I}(X; Y_1, Y_{2:K})$. Now assume a witness function: $f(x, y_{1:K}) = 1 + \log \frac{e^{f(x,y_1)}}{a(x,y_{1:K})}$, where $a(x, y_{1:K}) = \frac{1}{K}\sum_{i=1}^{K} e^{f(x,y_i)}$, a Monte Carlo approximation of the partition $Z(y)$. Applying the Nguyen-Wainwright-Jordan bound:

$$\begin{aligned}
\mathcal{I}(X; Y_1) &= \mathcal{I}(X; Y_1, Y_{2:K}) \\
&\geq \mathbb{E}_{p(x,y_{1:K})}\left[1 + \log \frac{e^{f(x,y_1)}}{a(x, y_{1:K})}\right] - \mathbb{E}_{p(x)p(y_{1:K})}\left[\frac{1}{e} \cdot e^{1+\log \frac{e^{f(x,y_1)}}{a(x,y_{1:K})}}\right] \\
&= 1 + \mathbb{E}_{p(x,y_{1:K})}\left[\log \frac{e^{f(x,y_1)}}{a(x, y_{1:K})}\right] - \mathbb{E}_{p(x)p(y_{1:K})}\left[\frac{e^{f(x,y_1)}}{a(x, y_{1:K})}\right]
\end{aligned} \tag{16}$$

Because $p(y_{1:K}) = \prod_i p(y_i)$ is invariant under permutation of indices $1 : K$, we have $\mathbb{E}_{p(x)p(y_{1:K})}\left[\frac{e^{f(x,y_1)}}{a(x,y_{1:K})}\right] = \mathbb{E}_{p(x)p(y_{1:K})}\left[\frac{e^{f(x,y_i)}}{a(x,y_{1:K})}\right]$ for any i. Using this and the definition of a:

$$\mathbb{E}_{p(x)p(y_{1:K})}\left[\frac{e^{f(x,y_1)}}{a(x, y_{1:K})}\right] = \frac{1}{K}\sum_{i=1}^{K} \mathbb{E}_{p(x)p(y_{1:K})}\left[\frac{e^{f(x,y_i)}}{a(x, y_{1:K})}\right] = \mathbb{E}_{p(x)p(y_{1:K})}\left[\frac{\frac{1}{K}\sum_{i=1}^{K} e^{f(x,y_i)}}{a(x, y_{1:K})}\right] = 1$$

We now substitute this result into the last term of Eq. 16:

$$\begin{aligned}
\mathcal{I}(X;Y_1) &\geq 1 + \mathbb{E}_{p(x,y_{1:K})}\left[\log \frac{e^{f(x,y_1)}}{a(x,y_{1:K})}\right] - 1 \\
&= \mathbb{E}_{p(x,y_{1:K})}\left[f(x,y_1) - \log \frac{1}{K}\sum_{i=1}^{K} e^{f(x,y_i)}\right] \\
&= \mathbb{E}_{p(x,y_1)}\mathbb{E}_{p(y_{2:K})}\left[f(x,y_1) - \log \frac{1}{K}\sum_{i=1}^{K} e^{f(x,y_i)}\right] \\
&= \mathbb{E}_{p(x,y_1)}\left[f(x,y_1) - \mathbb{E}_{p(y_{2:K})}\left[\log \frac{1}{K}\sum_{i=1}^{K} e^{f(x,y_i)}\right]\right] = \mathcal{I}^{\text{NCE}}(X;Y_1)
\end{aligned}$$

□

B Proof of VINCE

Proof. We start with the InfoNCE bound (Eq. 16), which similarly uses auxiliary variables.

$$\mathcal{I}(X;Y_1) \geq 1 + \mathbb{E}_{p(x|y_1)p(y_1)p(y_{2:K})}\left[\log \frac{e^{f(x,y_1)}}{a(x,y_{1:K})}\right] - \mathbb{E}_{p(x)p(y_1)p(y_{2:K})}\left[\frac{e^{f(x,y_1)}}{a(x,y_{1:K})}\right]$$

Fix $a(x,y_{1:K}) = \frac{1}{K}\sum_{i=1}^{K} e^{f(x,y_i)}$. Consider the second expectation only (call this $\mathcal{L}$):

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{p(x)p(y_{1:K})}\left[\frac{e^{f(x,y_1)}}{a(x,y_{1:K})}\right] \\
&= \mathbb{E}_{p(x)p(y_1)}\left[\int_{y_2} p(y_2)\cdots\int_{y_K} p(y_K)\left(\frac{e^{f(x,y_1)}}{a(x,y_{1:K})}\right) dy_K \cdots dy_2\right] \\
&= \mathbb{E}_{p(x)p(y_1)}\left[\int_{y_2} q(y_2|y_1)\cdots\int_{y_K} q(y_K|y_1)\left(\frac{e^{f(x,y_1)}}{a(x,y_{1:K})}\prod_{j=2}^{K}\frac{p(y_j)}{q(y_j|y_1)}\right) dy_K \cdots dy_2\right] \\
&= \mathbb{E}_{p(x)p(y_1)q(y_{2:K}|y_1)}\left[\frac{e^{f(x,y_1)}}{a(x,y_{1:K})}\prod_{j=2}^{K}\frac{p(y_j)}{q(y_j|y_1)}\right]
\end{aligned}$$

In changing the expectation, we are implicitly assuming that the supports for p and q are identical. Otherwise, the equality between integrals may not hold.

Next, we observe the following property of permutation invariance. Consider a bijection between two copies of the natural numbers between 1 and K inclusive. For example, if $K = 3$ consider $(1,2,3)$ and $(3,1,2)$. Fix an arbitrary bijection between $(1,\ldots K)$ and $(i_1,\ldots,i_K)$. Then, we note that $a(x,y_{1:K}) = a(x,y_{i_{1:K}})$ and $p(y_1)q(y_{2:K}|y_1) = p(y_{i_1})q(y_{i_{2:K}}|y_{i_1})$. For notational ease, we fix $q(y_{1:K}) =$

$p(y_i) \prod_{j \neq i} q(y_j|y_i)$ for *any* bijection mapping 1 to i. Consequently,

$$\mathbb{E}_{p(x)p(y_1)q(y_{2:K}|y_1)} \left[\frac{e^{f(x,y_1)}}{a(x,y_{1:K})} \prod_{j=2}^{K} \frac{p(y_j)}{q(y_j|y_1)} \right] = \mathbb{E}_{p(x)p(y_{i_1})q(y_{i_{2:K}}|y_{i_1})} \left[\frac{e^{f(x,y_{i_1})}}{a(x,y_{i_{1:K}})} \prod_{j=2}^{K} \frac{p(y_{i_j})}{q(y_{i_j}|y_{i_1})} \right]$$

If we take K such bijections and average over the expectation term for each bijection, we find

$$\begin{aligned}
\mathcal{L} &= \frac{1}{K} \sum_{i=1}^{K} \left(\mathbb{E}_{p(x)q(y_{1:K})} \left[\prod_{j \neq i} \frac{p(y_j)}{q(y_j|y_i)} \left(\frac{e^{f(x,y_i)}}{a(x,y_{1:K})} \right) \right] \right) \\
&= \mathbb{E}_{p(x)q(y_{1:K})} \left[\frac{\frac{1}{K} \sum_{i=1}^{K} \left(e^{f(x,y_i)} \prod_{j \neq i} \frac{p(y_j)}{q(y_j|y_i)} \right)}{\frac{1}{K} \sum_{i=1}^{K} e^{f(x,y_i)}} \right] \\
&\leq \mathbb{E}_{p(x)q(y_{1:K})} \left[\frac{\left(\sum_{i=1}^{K} e^{f(x,y_i)} \right) \left(\sum_{i=1}^{K} \prod_{j \neq i} \frac{p(y_j)}{q(y_j|y_i)} \right)}{\sum_{i=1}^{K} e^{f(x,y_i)}} \right] \\
&= K \mathbb{E}_{p(x)q(y_{1:K})} \left[\frac{1}{K} \sum_{i=1}^{K} \prod_{j \neq i} \frac{p(y_j)}{q(y_j|y_i)} \right] = K \mathbb{E}_{p(x)p(y_1)q(y_{2:K}|y_1)} \left[\prod_{j=2}^{K} \frac{p(y_j)}{q(y_j|y_1)} \right] = K
\end{aligned}$$

where the inequality holds since all elements in the both sums are positive. In the last line, we again apply a permutation invariance argument (but in the opposing direction). We reach the final equality since $\mathbb{E}_{p(x)p(y_1)q(y_{2:K}|y_1)} \left[\prod_{j=2}^{K} \frac{p(y_j)}{q(y_j|y_1)} \right] = \mathbb{E}_{p(x)p(y_1)p(y_{2:K})}[1]$. Next, consider the full expression,

$$\begin{aligned}
\mathcal{I}(X;Y_1) &\geq 1 + \mathbb{E}_{p(x|y_1)p(y_1)p(y_{2:K})} \left[\log \frac{e^{f(x,y_1)}}{a(x,y_{1:K})} \right] - \mathcal{L} \\
&\geq 1 + \mathbb{E}_{p(x,y_1)q(y_{2:K}|y_1)} \left[\prod_{j=2}^{K} \frac{p(y_j)}{q(y_j|y_1)} \left(\log \frac{e^{f_{\theta,\phi}(x,y_1)}}{a(x,y_{1:K})} \right) \right] - K = \mathcal{I}^{\text{VINCE}}(X;Y_1)
\end{aligned}$$

As a final note, we do not enforce that q be conditioned on y_1. The proof holds if $q(y|y_1) = q(y)$. □

C Learning a Proposal Distribution

Imagine we are optimizing the VINCE objective with respect to parameters θ and ϕ. As presented thus far, the proposal distribution is fixed i.e. it does not change over the course of optimization. However, in a truly variational sense, we may find that we wish to parameterize and learn the proposal distribution. For instance, we may find that easier negative samples are more efficient for learning early in optimization whereas more difficult negative samples are required late in optimization. While our family of allowed proposals does not support arbitrary learned distributions, we can parameterize and learn an acceptance probability function assuming a rejection sampled proposal distribution. Concretely, given a fixed sample $y_1 \sim p(y_1)$, define a parameterized proposal distribution $q_\psi(y_j|y_1) = r_\psi(y_j|y_1)p(y_j)$ for $j = 2, \ldots, K$ where $r_\psi(y_j|y_1) \in (0, 1]$ is an acceptance probability function. Since Corollary 3.1.1 supports an arbitrary acceptance probability function, for any choice of ψ, the resulting VINCE estimator is a lower bound on mutual information. We call this estimator with a learned acceptance probabiliy function, VINCE+. Buildng on Eq. 4, the VINCE+ objective (ignoring constants) is

$$\mathcal{I}^{\text{VINCE+}} = \mathbb{E}_{p(x,y_1)}\left[\left(\prod_{j=2}^{K}\frac{1}{r_\psi(y_j|y_1)}\right)\left(f_{\theta,\phi}(x,y_1) - \mathbb{E}_{q_\psi(y_{2:K}|y_1)}\left[\log\frac{1}{K}\sum_{i=1}^{K}e^{f_{\theta,\phi}(x,y_i)}\right]\right)\right] \tag{17}$$

During optimization, we must take gradients of Eq. 17 with respect to ψ, θ, and ϕ. The latter two are straightforward using modern auto-differentiation techniques. The first, however, is not as obvious as it requires taking the gradient of an expectation with respect to a rejection sampled distribution (one without closed form). We show in the next Lemma how to compute $\nabla_\psi \mathcal{I}^{\text{VINCE+}}$.

Lemma C.1. *Fix $y_1 \sim p(y_1)$. Define $q_\psi(y_j|y_1) = r_\psi(y_j|y_1)p(y_j)$ where $r_\psi(y_j|y_1) \in (0, 1]$. The gradient $\nabla_\psi \mathcal{I}^{\text{VINCE+}}$ can be computed as the following:*

$$\nabla_\psi \mathcal{I}^{\text{VINCE+}} = \mathbb{E}_{p(x,y_1)}[\nabla_\psi \mathcal{L}_1 - \nabla_\psi \mathcal{L}_2 - \nabla_\psi \mathcal{L}_3]$$

where (dropping θ and ϕ for clarity):

$$\nabla_\psi \mathcal{L}_1 = -f(x,y_1)\left(\sum_{i=2}^{K}\frac{1}{r_\psi(y_i|y_1)^2}\left(\prod_{j\neq i}\frac{1}{r_\psi(y_j|y_1)}\right)\right)$$

$$\nabla_\psi \mathcal{L}_2 = \mathbb{E}_{q_\psi(y_{2:K}|y_1)}\left[\left(\prod_{j=2}^{K}\frac{1}{r_\psi(y_j|y_1)}\right)\left(\log\frac{1}{K}\sum_{i=1}^{K}e^{f(x,y_i)}\right)\left(\nabla_\psi\sum_{j=2}^{K}\log r_\psi(y_j|y_1)\right)\right]$$

$$\nabla_\psi \mathcal{L}_3 = -\mathbb{E}_{q_\psi(y_{2:K}|y_1)}\left[\left(\sum_{i=2}^{K}\frac{1}{r_\psi(y_i|y_1)^2}\left(\prod_{j\neq i}\frac{1}{r_\psi(y_j|y_1)}\right)\nabla_\psi r_\psi(y_i|y_1)\right)\left(\log\frac{1}{K}\sum_{i=1}^{K}e^{f(x,y_i)}\right)\right]$$

Proof. Expand the expectation and use the score function estimator trick. Define $h_\psi(x, y_{1:K}) = \left(\prod_{j=2}^{K}\frac{1}{r_\psi(y_j|y_1)}\right)\left(\log\frac{1}{K}\sum_{i=1}^{K}e^{f_{\theta,\phi}(x,y_i)}\right)$. We consider each loss component individually:

$$\begin{aligned}\nabla_\psi \mathcal{L}_1 &= \nabla_\psi f_{\theta,\phi}(x,y_1)\cdot\left(\prod_{j=2}^{K}\frac{1}{r_\psi(y_j|y_1)}\right) + f_{\theta,\phi}(x,y_1)\cdot\nabla_\psi\left(\prod_{j=2}^{K}\frac{1}{r_\psi(y_j|y_1)}\right)\\ &= f(x,y_1)\left(\sum_{i=2}^{K}\frac{-1}{r_\psi(y_i|y_1)^2}\left(\prod_{j\neq i}\frac{1}{r_\psi(y_j|y_1)}\right)\right)\end{aligned}$$

Consider the remaining terms in Eq. 17.

$$\begin{aligned}
\nabla_\psi(\mathcal{L}_2 + \mathcal{L}_3) &= \nabla_\psi \mathbb{E}_{q_\psi(y_{2:K}|y_1)}\left[h_\psi(y_{1:K})\right] \\
&= \int_{y_{2:K}} \nabla_\psi (q_\psi(y_{2:K}|y_1) h_\psi(x, y_{1:K})) dy_{2:K} \\
&= \int_{y_{2:K}} h_\psi(x, y_{1:K}) \nabla_\psi q_\psi(y_{2:K}|y_1) dy_{2:K} + \int_{y_{2:K}} q_\psi(y_{2:K}|y_1) \nabla_\psi h_\psi(x, y_{1:K}) dy_{2:K} \\
&= \nabla_\psi \mathcal{L}_2 + \nabla_\psi \mathcal{L}_3
\end{aligned}$$

We employ the following transformation

$$\begin{aligned}
\nabla_\psi q_\psi(y_{2:K}|y_1) &= q_\psi(y_{2:K}|y_1) \nabla_\psi \log q_\psi(y_{2:K}|y_1) \\
&= q_\psi(y_{2:K}|y_1) \nabla_\psi (\log r_\psi(y_{2:K}|y_1) + \log p(y_{2:K})) \\
&= q_\psi(y_{2:K}|y_1) \nabla_\psi \log r_\psi(y_{2:K}|y_1)
\end{aligned}$$

We can simplify $\nabla_\psi \mathcal{L}_2$ and $\nabla_\psi \mathcal{L}_3$ further.

$$\begin{aligned}
\nabla_\psi \mathcal{L}_2 &= \int_{y_{2:K}} h_\psi(x, y_{1:K}) \nabla_\psi q_\psi(y_{2:K}|y_1) dy_{2:K} \\
&= \int_{y_{2:K}} h_\psi(x, y_{1:K}) q_\psi(y_{2:K}|y_1) \nabla_\psi \log r_\psi(y_{2:K}|y_1) dy_{2:K} \\
&= \mathbb{E}_{q_\psi(y_{2:K}|y_1)}[h_\psi(x, y_{1:K}) \nabla_\psi \log r_\psi(y_{2:K}|y_1)] \\
&= \mathbb{E}_{q_\psi(y_{2:K}|y_1)}[h_\psi(x, y_{1:K}) \nabla_\psi \sum_{j=2}^{K} \log r_\psi(y_j|y_1)]
\end{aligned}$$

where $r_\psi(y_{2:K}|y_1) = \prod_{j=2}^{K} r_\psi(y_j|y_1)$. For the remaining term,

$$\begin{aligned}
\nabla_\psi \mathcal{L}_3 &= \int_{y_{2:K}} q_\psi(y_{2:K}|y_1) \nabla_\psi h_\psi(x, y_{1:K}) dy_{2:K} = \mathbb{E}_{q_\psi(y_{2:K}|y_1)}[\nabla_\psi h_\psi(x, y_{1:K})] \\
&= \mathbb{E}_{q_\psi(y_{2:K}|y_1)}\left[\left(\sum_{i=2}^{K} \frac{-1}{r_\psi(y_i|y_1)^2}\left(\prod_{j\neq i} \frac{1}{r_\psi(y_j|y_1)}\right) \nabla_\psi r_\psi(y_i|y_1)\right)\left(\log \frac{1}{K}\sum_{i=1}^{K} e^{f(x,y_i)}\right)\right]
\end{aligned}$$

which just follows by chain rule. □

Lemma C.1 can be computed efficiently with auto-differentiation as long as r_ψ is differentiable.

D ImageNet Experiment Details

Unless otherwise specified, we follow [26]: when training on ImageNet, we used a temperature $\tau = 0.07$ and memory bank update rate $t = 0.5$. For optimization, we used SGD with momentum of 0.9, batch size 256, weight decay of 1e-4, learning rate 0.03. For pretraining, the learning rate was dropped twice by a factor of 10 once learning “converged", defined as no improvment in validation loss for 10 epochs. To define the background neighbors, we used 4096 of the nearest points. To define the close neighbors, we used 10 kNN with different initializations where $k = 30000$. To train kNNs, we use the FAISS libary [11]. For the image model architecture, we used ResNet18 [9] without pretraining where the number of classes is set to 128, our desired representation dimension. Again following previous work, for image augmentations during training, we use a random resized crop to 224 by 224, random grayscale with probability 0.2, random color jitter, random horizontal flip, and normalize pixels using ImageNet statistics. For testing, we resize to 256 by 256, center crop to 224 by 224 and normalize. When training LA, we initialize ResNet and the memory bank using 10 epochs of IR. We recommend the reader refer to [24, 26] for more details.