
Investigating the Role of Negatives in Contrastive Representation Learning

Jordan T. Ash*

Surbhi Goel*

Akshay Krishnamurthy*

Dipendra Misra*

Microsoft Research New York

(* all authors contributed equally)

Abstract

Noise contrastive learning is a popular technique for unsupervised representation learning. In this approach, a representation is obtained via reduction to supervised learning, where given a notion of semantic similarity, the learner tries to distinguish a similar (positive) example from a collection of random (negative) examples. The success of modern contrastive learning pipelines relies on many design decisions, such as the choice of data augmentation, the number of negative examples, and the batch size; however, there is limited understanding as to how these parameters interact and affect downstream performance. We focus on disambiguating the role of one of these parameters: the number of negative examples. Theoretically, we show the existence of a collision-coverage trade-off suggesting that the optimal number of negative examples should scale with the number of underlying concepts in the data. Empirically, we scrutinize the role of the number of negatives in both NLP and vision tasks.

1 INTRODUCTION

Unsupervised representation learning refers to a suite of algorithmic approaches for discovering meaningful transformations of complex data in the absence of an explicit supervision signal. These representations are trained to preserve and expose semantic information in the data, so that once learned, they can be used effectively in downstream tasks of interest. In recent years, these methods have seen tremendous success when

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

combined with deep learning models, obtaining state of the art results on a variety of important language and vision tasks (Smith and Eisner, 2005; Chopra et al., 2005; Mikolov et al., 2013; Oord et al., 2018; Bachman et al., 2019; Clark et al., 2020; Chen et al., 2020).

One popular approach for representation learning is contrastive learning, or noise contrastive estimation (NCE) (Gutmann and Hyvärinen, 2010). In NCE, a notion of semantic similarity is used to create a *self-supervision* signal that encourages similar points to be represented (or embedded) in a similar manner. For example, a typical approach is to train an embedding so that similar points have large inner product in the embedding space and dis-similar points have small inner product. Intuitively, representations trained in this manner capture the semantic similarity of the data, so we might expect that they are useful in downstream supervised learning tasks. Indeed, NCE is employed in the ELECTRA language model (Clark et al., 2020) and has also been used effectively in computer vision tasks at ImageNet scale (Chen et al., 2020).

While the implementation details vary, a mathematical abstraction for representation learning with NCE is: (1) a single NCE example consists of $k + 2$ raw inputs, $(x, x^+, x_1^-, \dots, x_k^-)$, where (x, x^+) are semantically similar and x_i^- are sampled from some base distribution, (2) the representation f is trained to encourage $f(x)^\top f(x^+) \gg f(x)^\top f(x_i^-)$ for each i . This second step can be done with standard classification objectives, such as the cross-entropy loss, where the model is viewed as a classifier over $k + 1$ labels. For example, a candidate objective is

$$\mathbb{E}_{x, x^+, x_{1:k}^-} \left[-\log \left(\frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^k e^{f(x)^\top f(x_i^-)}} \right) \right].$$

This objective encourages $f(x)^\top f(x^+)$ to be large, so that the representation efficiently captures the semantic similarity inherent in the NCE sampling scheme.

Modern contrastive pipelines have several parameters that together contribute to the success of resulting

representations. These parameters include modeling choices, such as the number of negatives k , the semantic sampling procedure for (x, x^+) , and neural network architecture, along with optimization decisions, such as batch size. While there have been large scale empirical studies on contrastive learning (Chen et al., 2020), it remains unclear how these different choices interact with each other to affect the downstream performance of the learned representations. Indeed, a major challenge is that exorbitant computational costs for training contrastive models precludes comprehensive ablations to disentangle the effects of each design decision. Instead a hybrid approach that leverages theoretical insights to guide systematic experimentation may be more effective at shedding light on the role of these parameters.

As an example of entangled design choices, consider the number of negative samples k and the batch size B in the NCE objective. In practice, it is standard to set k as a function of the batch size B , as $k = 2B - 2$, and then to set the batch size as large as the hardware can support. When implemented on a GPU, this allows for many more gradient evaluations with minimal computational overhead. However, in this setup, the effect of increasing k on downstream performance is confounded by improvements from increasing the batch size.

In this paper, we focus on the number of negatives k , and investigate its role in contrastive learning. The key question we ask is *how does the number of negative samples affect the quality of representations learned by NCE?* As mentioned, most empirical evidence confounds k with the batch size B , so it does not provide much of an answer to this question. Moreover, conceptual reasoning paints a fairly nuanced picture. On one hand, a large value of k implies that the negatives are more likely to cover the semantic space (*coverage*), which encourages the representation to disambiguate all semantically distinct concepts. However, there is a risk that some of the negative samples will be semantically very similar (*collisions*) to the positive example x^+ , which compromises the supervision signal and may lead to worse representation quality.

Our contributions. For our theoretical results, we adopt the framework of Saunshi et al. (2019), where the semantic similarity in the contrastive distribution is directly related to the labels in a downstream classification task. We refine the results of Saunshi et al. (2019) with a novel and tighter *error transfer* theorem that bounds the classification error by a function of the NCE error. We then study how this function depends on the number of concepts/classes in the data and the number of negatives in the NCE task. Our analysis captures the trade-off between coverage over the classes and collisions, and it reveals an expression for the ideal number of negatives k , which roughly scales as the

number of concepts in the data. We emphasize that our sharper theoretical results are instrumental even for obtaining this qualitative trend; indeed, the analysis in Saunshi et al. (2019) instead suggests that increasing k leads to decreasing performance.

Empirically, we study the effect of the number of negative samples on the performance of the final classifier on a downstream classification task for both vision and natural language datasets. For our language task, we observe that performance increases with k up to a point, but it deteriorates as we increase k further. The experiments also confirm that the optimal choice of k depends on the number of concepts in the dataset. In vision experiments, while we do not see performance degrade with increasing k , we do see that in general, performance does not continue to improve with increasing k .

2 RELATED WORK

Self-supervised learning has elicited breakthrough successes in representation learning, achieving state of the art performance across a wide array of important machine learning tasks, including computer vision (He et al., 2020; Huo et al., 2020; Khosla et al., 2020; Wang et al., 2020; Bachman et al., 2019), natural language processing (Fang and Xie, 2020; Fu et al., 2021; Le-Khac et al., 2020; Giorgi et al., 2020), and deep reinforcement learning. In vision, Chen et al. (2020) have shown that simple NCE algorithms are able to obtain competitive performance on the challenging ImageNet dataset, something that was previously thought to require extensive supervised training. In NLP, the NCE approach has garnered much attention since the pioneering work of Mikolov et al. (2013) demonstrated its efficacy in learning word-level representations. In Deep RL, NCE-style losses have been recently employed to learn world models in complex environments (Nachum et al., 2018; Laskin et al., 2020; Zhu et al., 2020; Kipf et al., 2019; Shelhamer et al., 2017).

Theoretical work on representation learning is quite nascent, with most works focused on mathematical explanations for the efficacy of these approaches (Saunshi et al., 2019; Lee et al., 2020). One is via conditional independence and redundancy (Tosh et al., 2020, 2021). Another theory proposed by Wang and Isola (2020) is that contrastive representations simultaneously optimize for alignment of similar points and uniformity in the embedding space, in the asymptotic regime where the number of negatives tends to infinity. This asymptotic regime was also studied by Foster et al. (2020), who analyze a contrastive estimator for an information-gain quantity. However, Foster et al. (2020) do not study representation learning and Wang and Isola (2020) do not theoretically analyze downstream performance as we do. Indeed, our results show

that there can be a price for too many negatives due to collisions, and perhaps call into question whether this asymptotic regime should be used to explain the downstream performance of NCE representations.¹

Various issues associated with negative sampling, including collisions, have been identified in prior empirical works (Kalantidis et al., 2020; Huynh et al., 2022), although the approaches proposed to address these issues have limited theoretical justification. One exception is the work of Chuang et al. (2020), who propose a debiased version of the contrastive loss that attempts to correct for negative samples coming from the positive class. They experimentally observe significant gains from removing the effect of collision.

In contrast to these works, we study the original contrastive objective that is primarily used in practice from a theoretical perspective and characterize how an appropriate choice of k can balance the collision and coverage trade-off. Concurrently, Nozawa and Sato (2021) also study the effect of increasing k using similar techniques (specifically, the coupon collector). However, their analysis holds in the regime where k is larger than the number of underlying concepts, unlike ours, which shows a trade-off at smaller k .

Our theoretical results closely follow the setup of Saunshi et al. (2019). We adopt essentially the same framework in the present paper, but unlike Saunshi et al. (2019) we do not use the number of negatives k to define the downstream classification tasks. Our setup yields a single k -independent downstream task, which allows us to study the influence of this parameter. Further, many steps of our analysis are sharper than theirs, leading to new qualitative findings.

Finally, we mention that recent work proposes representation learning pipelines that do not use negative samples at all (Grill et al., 2020; Chen and He, 2021). It is an interesting question to understand the design decisions in these pipelines (Tian et al., 2021), but this is outside the scope of the present paper.

3 SETUP AND ALGORITHM

In this section, we present the mathematical framework that we use to study NCE, as well as the learning algorithm and the evaluation metric that we consider.

Notation. Let the set of input features be denoted by \mathcal{X} and the set of latent classes by \mathcal{C} , where $|\mathcal{C}| = N$. With each latent class $c \in \mathcal{C}$ we will associate a distribution \mathcal{D}_c over \mathcal{X} . We can view \mathcal{D}_c as the distribution over data conditioned on belonging to latent class c .

¹Note that similar concerns have been raised regarding viewing representation learning as a mutual information maximization problem (Tschannen et al., 2019).

We will also assume a distribution ρ on \mathcal{C} . We will learn with a function class \mathcal{F} , the set of representation functions $f : \mathcal{X} \rightarrow \mathbb{R}^d$ that map the input to a d -dimensional vector. We use \lesssim to denote less than or equal to up to constants.

NCE data. We assume access to *similar* data points in the form of pairs (x, x^+) and k *negative* data points x_1^-, \dots, x_k^- . To formalize this, an unlabeled sample from \mathcal{D}_{NCE} is generated as follows:

- Choose a class c according to ρ and draw i.i.d. samples x, x^+ from \mathcal{D}_c .
- Choose classes c_1^-, \dots, c_k^- i.i.d. from ρ and draw $x_1^- \sim \mathcal{D}_{c_1^-}, \dots, x_k^- \sim \mathcal{D}_{c_k^-}$.
- Output tuple $(x, x^+, x_{1:k}^-)$

This data generation is implicit. We only observe the data points and not the associated class labels. Observe that c_i^- may be the same as c for one or more $i \in \{1, \dots, k\}$, causing *collisions*.

Empirically, similar pairs are typically generated from co-occurring data, for example, two parts of a document can be viewed as a similar pair (Tosh et al., 2020). Note that our data generation process implies that x and x^+ are independent conditioned on the latent class. This is a simplification of the real-world setting where independence might not hold. However, it enables a transparent theoretical analysis that can provide insights that are applicable to some real-world setups.

NCE algorithm. The objective of an NCE algorithm is to use the available unsupervised data to learn a representation that represents (or embeds) similar points in a similar manner and distinguishes them from the negative samples. The following NCE loss captures this objective,

Definition 1 (NCE Loss). *The NCE loss for a representation f on the distribution \mathcal{D}_{NCE} is defined as*

$$\mathcal{L}_{\text{NCE}}^{(k)}(f) := \mathbb{E}_{\mathcal{D}_{\text{NCE}}} \left[\ell \left(\left\{ f(x)^T (f(x^+) - f(x_i^-)) \right\}_{i=1}^k \right) \right].$$

The empirical NCE loss with a set S of M samples drawn from \mathcal{D}_{NCE} is $\widehat{\mathcal{L}}_{\text{NCE}}^{(k)}(f; S) := \frac{1}{M} \sum_{(x, x^+, x_{1:k}^-) \in S} \ell \left(\left\{ f(x)^T (f(x^+) - f(x_i^-)) \right\}_{i=1}^k \right)$.

We restrict ℓ to be one of the two standard loss functions, hinge loss $\ell(v) = \max\{0, 1 + \max_i\{-v_i\}\}$ and logistic loss $\ell(v) = \log(1 + \sum \exp(-v_i))$ for a vector $v \in \mathbb{R}^k$. Both loss functions and their variants are deployed routinely in practical NCE implementations (Schroff et al., 2015; Chen et al., 2020).

The algorithm we analyze is the standard empirical risk minimization (ERM) on $\widehat{\mathcal{L}}_{\text{NCE}}^{(k)}$ over \mathcal{F} . For the rest of the paper, we will denote our learned representation $\widehat{f} \in \arg \min_{f \in \mathcal{F}} \widehat{\mathcal{L}}_{\text{NCE}}^{(k)}(f)$

Downstream supervised learning task. For evaluating representations, we will consider the standard supervised learning task of classifying a data point into one of the classes in \mathcal{C} . More formally, a sample from \mathcal{D}_{sup} is generated as follows: (1) choose a class c according to ρ (2) draw sample x from \mathcal{D}_c , and (3) output sample (x, c) .

Observe that the distribution over classes and the conditional distribution over data points is the same as in the unsupervised data. This is crucial to be able to relate the NCE task to the downstream classification task. We will use the corresponding supervised loss for function $g : \mathcal{X} \rightarrow \mathbb{R}^N$,

$$\mathcal{L}_{\text{sup}}(g) = \mathbb{E}_{\mathcal{D}_{\text{sup}}} \left[\ell \left(\{g(x)_c - g(x)_{c'}\}_{c' \neq c} \right) \right]$$

Note that [Saunshi et al. \(2019\)](#) consider a different downstream task which depends on the number of negative samples in the NCE task. Here we disentangle the number of negatives used in NCE from the number of downstream classes which results in a more natural downstream task.

To evaluate the performance of the learned representation function \hat{f} , we will train a linear classifier $W \in \mathbb{R}^{N \times d}$ on top of the learned embedding and use it for the downstream task. We overload notation and define the supervised loss of an embedding f as

$$\mathcal{L}_{\text{sup}}(f) = \inf_{W \in \mathbb{R}^{N \times d}} \mathcal{L}_{\text{sup}}(Wf).$$

We define the loss of the mean classifier as $\mathcal{L}_{\text{sup}}^\mu(f) = \mathcal{L}_{\text{sup}}(W^\mu f)$ where W^μ is such that the c^{th} row is the mean $\mu_c = \mathbb{E}_{x \sim \mathcal{D}_c} [f(x)]$ of the representations of data from a fixed class c . Note that $\mathcal{L}_{\text{sup}}(f) \leq \mathcal{L}_{\text{sup}}^\mu(f)$ for all f . Our guarantees will hold for the loss of the mean classifier which will directly imply a bound on the overall supervised loss. Note that this analysis may be loose if the mean-classifier itself has poor downstream performance².

4 THEORETICAL OVERVIEW

In this section, we present our main theoretical insights and overview our analysis, which provides refined bounds on the supervised loss in terms of the NCE loss. Our bounds show an interesting trade-off between the number of negatives and the number of latent classes. For ease of presentation, bounds in the main paper assume that ρ is uniform over \mathcal{C} . Proofs and generalizations are deferred to [Appendix B](#).

In the supervised loss, data points are distinguished against all classes other than the class that the point

²[Saunshi et al. \(2019\)](#) present a simple example where the two losses are very different.

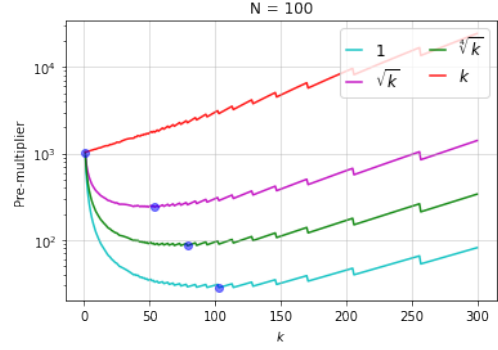


Figure 1: Value of pre-multiplier $\alpha(k, N)$ multiplied with the growth rates of $\mathcal{L}_{\text{NCE}}^{(k)} - \tau_k$ (on a log-scale) for $N = 100$ with varying k and growth rates. The plots represent the scaling of our upper bound on the supervised loss. The blue dots represent the optimal k predicted by our theory.

belongs to. However, for the NCE loss, if the number of negatives k is small relative to N , then the negatives may not cover all the classes simultaneously. On the other hand, if k is too large, then it is quite likely that some of the negatives will belong to the same class as the “anchor point” x , causing collisions that force the learner to distinguish between similar examples. This leads to opposing factors when choosing the optimal number of negative samples. We present our main result that captures this trade-off.

Theorem 1 (Transfer between NCE and supervised loss). *For any $f \in \mathcal{F}$,*

$$\begin{aligned} \mathcal{L}_{\text{sup}}(f) &\leq \mathcal{L}_{\text{sup}}^\mu(f) \\ &\leq \underbrace{\frac{2}{1 - \tau_k} \left[\frac{2(N-1)H_{N-1}}{k} \right]}_{\alpha(k, N)} \left(\mathcal{L}_{\text{NCE}}^{(k)}(f) - \tau_k \right), \end{aligned}$$

where $\tau_k = \Pr[\exists i : c_i^- = c] = 1 - \left(1 - \frac{1}{N}\right)^k$ is the probability of a collision amongst the negatives and $H_t = \Theta(\log(t))$ is the t^{th} harmonic number.

The above bound is similar to the one of [Saunshi et al. \(2019\)](#), except that the left hand side and the pre-multiplier $\alpha(k, N)$ are different. The left hand side is the supervised loss on all classes rather than the loss, in expectation, on $k + 1$ -way classification task, where tasks are drawn from ρ . Further, our pre-multiplier captures the trade-off between k and N . A key difference is that the bound in the prior work gets worse with increasing k implying $k = 1$ is the best choice whereas ours has a more subtle dependence on k .

Remark 2 (Choosing k). *The coefficient $\alpha(k, N)$ decreases with k up to $k \approx \frac{1}{\log(N) - \log(N-1)}$ and then increases. Note that $\mathcal{L}_{\text{NCE}} - \tau_k$ may itself increase as a function of k , however as long as its growth rate is $o(k)$, we will see a similar trend. (See [Figure 1](#)).*

One concern with the bound in [Theorem 1](#) is that it may not be possible to make $\mathcal{L}_{\text{NCE}}^{(k)}(f) - \tau_k$ small, and the minimum value may depend on k . This was also observed by [Saunshi et al. \(2019\)](#), who propose to decompose $\mathcal{L}_{\text{NCE}}^{(k)} - \tau_k$ into two terms, the NCE objective with no collisions and a measure of within-class deviation. Both of these terms can potentially be made small for large \mathcal{F} . The bound in [Theorem 1](#) can also be refined in a similar manner, although the proof requires some new arguments. We state the result next.

Theorem 3 (Refined NCE Bound). *For any $f \in \mathcal{F}$,*

$$\mathcal{L}_{\text{sup}}^\mu(f) \lesssim \alpha(k, N) \left(\mathcal{L}_{\neq}^{(k)}(f) + \sqrt{\frac{k}{N}} s(f) \right)$$

where $\mathcal{L}_{\neq}^{(k)}(f)$ is the NCE loss obtained by ignoring the coordinates corresponding to the collisions and $s(f)$ is a measure of average intra-class deviation.

We defer the formal definitions of $\mathcal{L}_{\neq}^{(k)}(f)$ and $s(f)$ to [Appendix B.2](#). Note that the trade-off between N and k still holds here even for the coefficient in front of $s(f)$ (see [Figure 1](#) for \sqrt{k}). $\alpha(k, N)\sqrt{\frac{k}{N}}$ is minimized roughly at $k \approx \frac{0.5}{\log(N) - \log(N-1)}$.

Lastly, we do a careful generalization analysis to guarantee closeness of the empirical NCE loss of \hat{f} to the true NCE loss. Given a generalization guarantee, using the fact that \hat{f} is the ERM solution, we obtain the following guarantee:

Theorem 4 (Guarantee on ERM). *For $\hat{f} \in \arg \min_{f \in \mathcal{F}} \hat{\mathcal{L}}_{\text{NCE}}^{(k)}(f)$ and any $f \in \mathcal{F}$,*

$$\mathcal{L}_{\text{sup}}(\hat{f}) \lesssim \alpha(k, N) \left(\mathcal{L}_{\neq}^{(k)}(f) + \sqrt{\frac{k}{N}} s(f) + \sqrt{k} \text{Gen}_M \right)$$

where Gen_M is a decreasing function of M that depends on \mathcal{F} but is independent of k .

See [Appendix B.3](#) for a formal definition of Gen_M , which is based on the Rademacher complexity of \mathcal{F} .

A coverage-collision trade-off example. Our bounds suggest that there are two competing forces influenced by the choice of k in the NCE task. The first is coverage of all the classes, which intuitively improves supervised performance, and suggests taking k larger. On the other hand, when k is large, the negatives are more likely to collide with the anchor class c , which will degrade performance. However, as the trends regarding the optimal choice of k are based on analyzing an upper bound, there is always the concern that the upper bound is loose. To further substantiate our results, we now provide a simple example which demonstrates the existence of the coverage-collision trade-off and also verifies the sharpness of our transfer theorem.

First we showcase the tightness of [Theorem 1](#) and the importance of coverage. Consider a problem with N classes $\{1, \dots, N\}$ where N is a multiple of 2 and let ρ be the uniform distribution. For each class c , the distribution D_c is uniform on two examples x_1^c and x_2^c . The representation f_1 pairs classes together, so that all points from class $2i$ and $2i + 1$ are both mapped to the i^{th} standard basis element e_i (see [Figure 2](#)).

It is straightforward to verify the following facts for the hinge loss (see the appendix for detailed calculations): (1) since f_1 collapses classes together, it has $\mathcal{L}_{\text{sup}}^\mu(f_1) = 1$, (2) $\mathcal{L}_{\text{NCE}}(f_1) = 1 - (1 - 2/N)^k$ since for each class $2i$ the NCE loss will be 1 if and only if either $2i$ or $2i + 1$ appear in the negatives (otherwise the loss will be 0). Thus, for this function f_1 we have

$$\frac{\mathcal{L}_{\text{NCE}}^{(k)}(f_1) - \tau_k}{1 - \tau_k} = 1 - \left(1 - \frac{1}{N-1}\right)^k \leq \frac{k}{N-1},$$

while $\mathcal{L}_{\text{sup}}^\mu(f_1) = 1$. This verifies that [Theorem 1](#) is tight up to constants and the $\Theta(\log(N-1))$ factor arising from the $N-1$ -st harmonic number.

In addition, observe that $\mathcal{L}_{\text{NCE}}^{(k)}(f_1)$ increases rapidly with k , since when k is large it is very likely that either the anchor class or its pair will appear as a negative. This demonstrates the importance of coverage: When k is large, f_1 , which has high downstream loss, is unlikely to be selected by the NCE procedure.

To make this precise, consider another function f_2 such that $f_2(x_1^i) = (1 + \epsilon)e_i$ and $f_2(x_2^i) = (1 - \epsilon)e_i$. This representation correctly separates all of the classes, but there is some noise in the embedding space. By direct calculation, we have $\mathcal{L}_{\text{sup}}^\mu(f_2) = \epsilon/2$ so that f_2 should be preferred for downstream performance. We can also show that,

$$0 \leq \mathcal{L}_{\text{NCE}}^{(k)}(f_2) - (1 - \tau_k) \left(\frac{\epsilon^2}{4} + \frac{\epsilon}{2} \right) - \tau_k \leq \tau_k \epsilon.$$

See [Figure 2](#), where we plot a sharper lower bound established in the appendix. For moderately small ϵ , say $\epsilon = 0.1$, these bounds imply that $\mathcal{L}_{\text{NCE}}^{(k)}(f_1) < \mathcal{L}_{\text{NCE}}^{(k)}(f_2)$ when k/N is small ($k \lesssim 0.05N$), which means that the bad representation f_1 will be selected. On the other hand when $k \gtrsim 0.05N$, the better representation f_2 will be selected, demonstrating how using many negatives can yield more performant representations.

In the asymptotic regime where $k \rightarrow \infty$, we observe a second important phenomena with this example: If k/N is too large, then the bad classifier f_1 will be favored again. To see this, note that when there are many collisions f_2 will incur $1 + \epsilon$ NCE error, so $\lim_{k \rightarrow \infty} \mathcal{L}_{\text{NCE}}^{(k)}(f_2) = 1 + \epsilon$. On the other hand, it is clear that $\lim_{k \rightarrow \infty} \mathcal{L}_{\text{NCE}}^{(k)}(f_1) = 1$, so f_1 will be selected. This phenomena can be seen in [Figure 2](#). In this

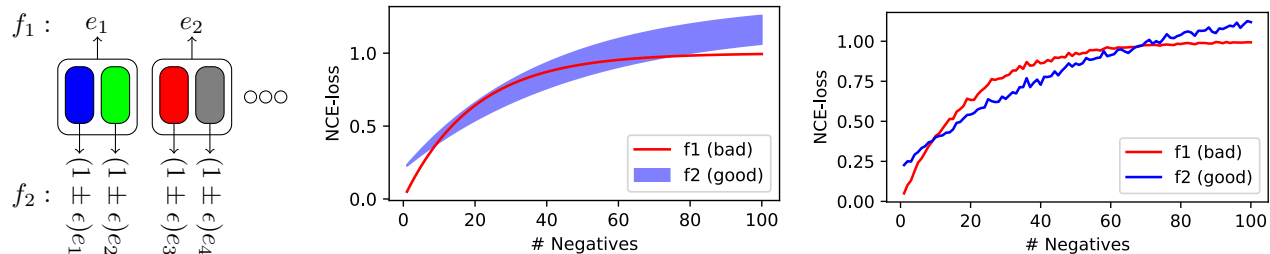


Figure 2: **Left:** Schematic of the distribution with the two representation functions, where colors correspond to distinct classes. f_1 collapses pairs of classes together, while f_2 separates all classes, but has some intraclass variance in its embeddings. **Center:** plot of $\mathcal{L}_{\text{NCE}}^{(k)}(f_1)$ and the bounds on $\mathcal{L}_{\text{NCE}}^{(k)}(f_2)$ as a function of k . **Right:** Monte-carlo simulations of $\mathcal{L}_{\text{NCE}}^{(k)}$ for the two representations, demonstrating both crossing points. In both plots we take $N = 40, \epsilon = 0.35$.

regime, the collisions in the NCE problem compromise the supervision signal and lead to the selection of a suboptimal representation.

5 EXPERIMENTS

We test our theoretical findings using on NLP and vision tasks. We first describe our general learning setting and then describe these tasks and results.

Experimental setting. We train the encoder model f by performing NCE with logistic loss. We train a linear classifier on the downstream task for a fixed number of epochs and do not fine-tune f during downstream training.

Computing k negative tokens for a batch size B results in $\mathcal{O}(Bk)$ forward passes through the model, which is prohibitively expensive for large values of B and k . In order to reduce computational overhead, we follow the data reuse trick used in literature (Chen et al., 2020; Chuang et al., 2020). Given B pairs of positively aligned, semantically similar points $\{(x_i, x_i^+)\}_{i=1}^B$, we compute $\{f(x_i), f(x_i^+)\}_{i=1}^B$ using $2B$ forward passes. For each $i \in [B]$, we treat the set $\mathcal{C}_i = \{x_j, x_j^+ \mid j \in [B], j \neq i\}$ as the candidate set of negative samples to pick from for the positive pair (x_i, x_i^+) . Previous work use the entire candidate set \mathcal{C}_i , which gives $k = |\mathcal{C}_i| = 2(B - 1)$. While this allows most negative samples to be used, it results in a confounding affect, due to the entanglement of B and k . Instead, for a fixed value of $k \leq 2(B - 1)$, we sample k negative examples without replacement from \mathcal{C}_i independently for each i . This sampling protocol marks a notable departure from prior work by decoupling the dependence between batch size and number of negative examples and studying their influence on performance separately.

NLP experiments. Here we perform a thorough analysis on the Wiki-3029 dataset of Saunshi et al. (2019). The data consist of 200 sentences taken from

3029 Wikipedia articles. The downstream task is to predict the article from which a given sentence was taken. A key advantage of these data are that they contain many classes, allowing us to study learning behavior as the number of classes N increases. For a given $N \leq 3029$, for both training and test datasets, we remove sentences from all but the first N articles. We tokenize sentences using the NLTK English word tokenizer (Bird et al., 2009) and replace all tokens that occur fewer than 50 times in the training set with UNK, denoting an *unknown token*. This gives us a vocabulary size V of 15,807 with at most 136 unique tokens per sentence. We create our training set by randomly sampling 80% of the sentences from each article. The remaining 20% of the sentences for each article form the test set.

We use a bag-of-words representation for modeling f . Formally, we map a sentence x to a list of tuples $\{(t_i, p_i)\}$ where $\{t_i\}$ is the set of unique tokens in x and p_i is the number of times t_i occurs in x normalized by the number of tokens in x . We compute $f(x) = \sum_i p_i \nu_{t_i}$ where $\nu \in \mathbb{R}^{V \times d}$ is a randomly initialized word-embedding matrix with embedding dimension d and ν_{t_i} is the word embedding of token t_i .

We train the NCE model for a fixed number of epochs and do not use a held-out set, following the protocol of Saunshi et al. (2019). We measure both the downstream performance of the mean classifier W^μ and a trained linear classifier. For the former, we use the entire training dataset to compute W^μ using the trained representation. For the latter, we use the entire training dataset, with labels, to fit a linear layer on top of the model f , and we do not fine-tune f . For each setting, we repeat the experiment 5 times with different seeds. See Appendix D for hyperparameter values and optimization details.

The leftmost plot in Figure 3 shows the accuracy of the mean classifier as we vary N and k . For a given number

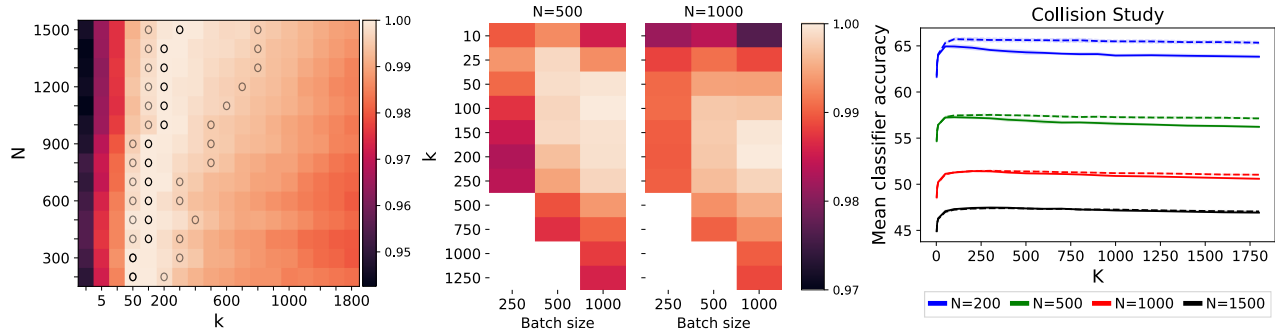


Figure 3: Mean classifier accuracy on the **Wiki-3029** dataset. **Left:** Downstream performance as a function of N and k for a fixed batch size of $B = 1000$. Each row is locally normalized by dividing the unnormalized accuracy in a given row with the best performance in that row. Color denotes normalized average accuracy across 5 trials where lighter areas have higher performance. Solid black circles highlight k with the best performance for each row. Shaded black circles show the smallest and largest value of k with normalized performance of at least 0.995. **Center:** Mean classifier average accuracy across 5 trials with varying B and K . The performance is globally normalized by dividing the unnormalized accuracy by the overall best accuracy. **Right:** An experiment highlighting the effect of disabling collisions across k for 4 values of N . Solid lines indicate vanilla training, and dotted lines indicate no collision.

of classes N , performance improves as k increases and then slowly decays for larger values of k . We show the general trend in this figure and report exact numbers to the appendix. Further, we find that the optimal choice of k , indicated by solid black circles, slowly increases as N increases. Both of these empirical findings support the theoretical predictions of our improved error transfer bounds. As we have discussed, the previous theoretical results (Saunshi et al., 2019) do not predict this observed trend. Lastly, we find that performance improves very rapidly as k increases, up until $k = 50$, while the decay in performance as k increases beyond the optimal value is mild in comparison to these initial gains.

The center plot in Figure 3 shows the mean classifier’s performance as we vary B and k for two values of N . Note that for a given value of B , we can only increase k up to $2(B - 1)$. We observe that increasing the batch size can improve the performance even when the number of negative examples are fixed. Optimal performance is obtained for the largest batch size and an intermediate value of k . This highlights the very different roles played by B and k on this dataset. We note that these results could not have been observed by previous work that entangle the value of k to B .

Our theoretical analysis reveals a coverage-collision trade-off and suggests that the poor performance when k is very large is due to many collisions between the negative samples and the anchor/positive. To empirically verify this claim, we modify the representation learning procedure by sampling k negative examples only from those members in the candidate set that do not share

the same class as the positive reference point.³ In the rightmost plot of Figure 3, we compare this sampling procedure with the standard one for different values of N and k and a fixed $B = 1000$. We find that artificially removing collisions improves the performance across all values of N ; however, the effect is more significant for smaller values of N (1.5% for $N = 200$ and $K = 1000$). As expected, this effect is more significant for small N ; the probability of collision decreases rapidly with N , so the different sampling procedures are quite similar for large N . We can further notice that performance decays slightly for very large values of k in the absence of collisions. This evidence supports the collision hypothesis, namely that collisions are a central reason for performance degradation with increasing k . Note that this observation was also made by Chuang et al. (2020). Lastly, as hypothesized by prior work (Chen et al., 2020), we believe a mild reduction in accuracy with large values of k can happen even in the absence of collision due to optimization instability.

Vision experiments. In this section, we decouple batch size from k and measure downstream performance as a function of these parameters on the SVHN and CIFAR-10 datasets. We fit all parameters except the last layer of a ResNet-18 model using the NCE loss, and we evaluate downstream accuracy by training the last layer only in a fully supervised fashion. As is done in SimCLR, we normalize these representations to have unit norm. We use a learning rate of 10^{-3} , train for 400 epochs, and fit parameters with the Adam optimizer.

³If we are unable to find k true negatives for any example in the batch, then no update is performed. However, due to the choice of N, B, k this never occurred in our experiments.

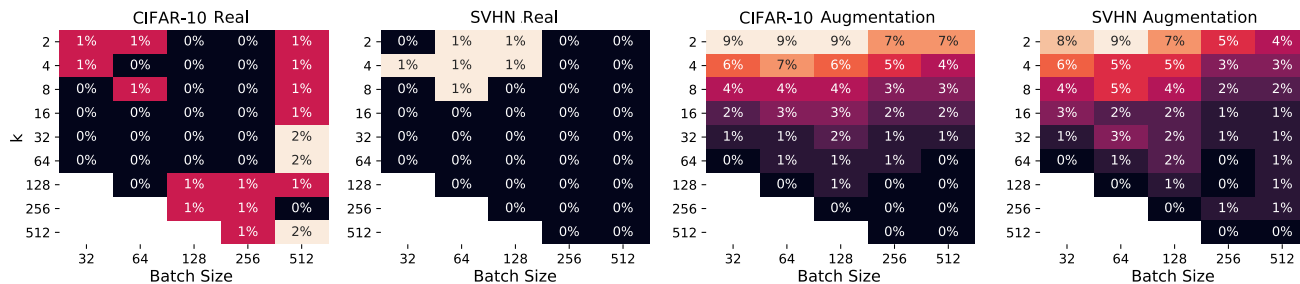


Figure 4: Average downstream classifier accuracy as a function of both batch size and k for various vision datasets and NCE parameters. **Left:** CIFAR-10 data, using a true positive sample from the same class, as used in our analysis. **Center Left:** SVHN data, also using a true positive. The rightmost figures are CIFAR-10 and SVHN respectively, where the positive sample is an augmented version of the reference sample, and where negatives are augmented as well. Accuracies are represented as a percent difference from the highest accuracy in the column.

We consider two ways of sampling tuples of images for the NCE objective. In the first, we adhere to our theoretical framework and use class information to capture semantic similarity. Here, for a given sample, we use an image from the same class to act as a positive, and we do not use data augmentation. These results are presented in Figure 4, where the accuracy of each k and batch size is presented as the percent difference between that cell and that of the best-performing k for that same batch size. For both datasets considered, performance does not seem to improve as k increases, and even relatively small k achieve peak performance. Unnormalized accuracies, as well as results using a mean classifier, can be found in Appendix D.

Figure 4 also presents experiments deviating from our theory: Instead of using true positives, we exercise the SimCLR data augmentation scheme, consisting of randomly applied crops, color jitter, horizontal flips, greyscaling, and blurring (Chen et al., 2020). In this setting, performance seems to improve with increasing k , although at a decreasing rate. Interestingly, recent empirical work examining the trade-off between batch size and k using the exact training protocol of Chen et al. (2020), shows a decay in performance with increasing k (Mitrovic et al., 2020). This approach differs from what is shown here primarily in that a two-layer MLP is appended to f to fit the NCE objective and discarded before downstream training.

6 DISCUSSION

In this paper, we study how the number of negative samples k used in contrastive representation learning affects the quality of learned representations. Our theoretical results highlight a coverage-collision trade-off and suggest that the number of negatives should scale roughly linearly with the number of concepts. Our NLP experiments demonstrate a degradation in performance with increasing k , while for vision experiments

performance generally stop improving as k increases. This article therefore raises a number of interesting directions for future work, spanning both theory and practice, and calls for a more detailed investigation to better understand the success of NCE in different domains. We close with open directions and related points of discussion:

Data augmentation. As alluded to by our vision experiments, the use of data augmentation in practice seems to be a principal factor in the gap between what is observed empirically and what we describe theoretically. Because NCE is so commonly used with image data, developing a theoretical model that can incorporate properties of commonly used augmentations and the inductive bias of neural networks is an important avenue for future research.

Choosing k in practice. When the NCE distribution and the downstream task are closely linked, our theoretical results indicate that the number of negatives used in NCE should scale with the number of classes. In practice, however, choosing k is less clear, with the optimal choice being a function of complex factors. The model’s inductive bias, the choice of data augmentation used, and the data type all seem to influence the ideal value. Given this, are there algorithmic principles for setting k in practice?

Decoupling negatives with other parameters. This work disentangles k from the batch size used for model training, with our experiments suggesting that each plays a different role in the quality of the learned representation. Modern NCE pipelines have several conflating parameters (such as choice of augmentations, model class, learning rate, stopping epoch), and we believe that a deeper empirical study is imperative for disambiguating and understanding the role of each.

References

- Bachman, P., Hjelm, R. D., and Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*.
- Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O’Reilly Media.
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*.
- Chen, X. and He, K. (2021). Exploring simple siamese representation learning. In *Conference on Computer Vision and Pattern Recognition*.
- Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. (2020). Debaised contrastive learning. In *Advances in Neural Information Processing Systems*.
- Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.
- Fang, H. and Xie, P. (2020). Cert: Contrastive self-supervised learning for language understanding. *arXiv:2005.12766*.
- Foster, A., Jankowiak, M., O’Meara, M., Teh, Y. W., and Rainforth, T. (2020). A unified stochastic gradient approach to designing bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*.
- Foster, D. J. and Rakhlin, A. (2019). ℓ_∞ -vector contraction for rademacher complexity. *arXiv:1911.06468*.
- Fu, H., Zhou, S., Yang, Q., Tang, J., Liu, G., Liu, K., and Li, X. (2021). Lrc-bert: Latent-representation contrastive knowledge distillation for natural language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Giorgi, J. M., Nitski, O., Bader, G. D., and Wang, B. (2020). Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv:2006.03659*.
- Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. (2020). Bootstrap your own latent—a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*.
- Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*.
- He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Huo, X., Xie, L., Wei, L., Zhang, X., Li, H., Yang, Z., Zhou, W., Li, H., and Tian, Q. (2020). Heterogeneous contrastive learning: Encoding spatial information for compact visual representations. *arXiv:2011.09941*.
- Huynh, T., Kornblith, S., Walter, M. R., Maire, M., and Khademi, M. (2022). Boosting contrastive self-supervised learning with false negative cancellation. In *Winter Conference on Applications of Computer Vision*.
- Kalantidis, Y., Sariyildiz, M. B., Pion, N., Weinzaepfel, P., and Larlus, D. (2020). Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*.
- Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning. In *Advances in Neural Information Processing Systems*.
- Kipf, T., van der Pol, E., and Welling, M. (2019). Contrastive learning of structured world models. In *International Conference on Learning Representations*.
- Laskin, M., Srinivas, A., and Abbeel, P. (2020). Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*.
- Le-Khac, P. H., Healy, G., and Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*.
- Lee, J. D., Lei, Q., Saunshi, N., and Zhuo, J. (2020). Predicting what you already know helps: Provable self-supervised learning. *arXiv:2008.01064*.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *International Conference on Learning Representations (Workshop Track)*.
- Mitrovic, J., McWilliams, B., and Rey, M. (2020). Less can be more in contrastive learning. In *“I Can’t Believe It’s Not Better!” NeurIPS 2020 workshop*.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.
- Nachum, O., Gu, S., Lee, H., and Levine, S. (2018). Near-optimal representation learning for hierarchical

- reinforcement learning. In *International Conference on Learning Representations*.
- Nozawa, K. and Sato, I. (2021). Understanding negative samples in instance discriminative self-supervised representation learning. *arXiv preprint arXiv:2102.06866*.
- Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv:1807.03748*.
- Saunshi, N., Plevrakis, O., Arora, S., Khodak, M., and Khandeparkar, H. (2019). A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*.
- Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- Shelhamer, E., Mahmoudieh, P., Argus, M., and Darrell, T. (2017). Loss is its own reward: Self-supervision for reinforcement learning. In *International Conference on Learning Representations (Workshop Track)*.
- Smith, N. A. and Eisner, J. (2005). Contrastive estimation: Training log-linear models on unlabeled data. In *Annual Meeting of the Association for Computational Linguistics*.
- Tian, Y., Chen, X., and Ganguli, S. (2021). Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*.
- Tosh, C., Krishnamurthy, A., and Hsu, D. (2020). Contrastive estimation reveals topic posterior information to linear models. *arXiv:2003.02234*.
- Tosh, C., Krishnamurthy, A., and Hsu, D. (2021). Contrastive learning, multi-view redundancy, and linear models. In *Algorithmic Learning Theory*.
- Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., and Lucic, M. (2019). On mutual information maximization for representation learning. In *International Conference on Learning Representations*.
- Wang, T. and Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*.
- Wang, X., Zhang, R., Shen, C., Kong, T., and Li, L. (2020). Dense contrastive learning for self-supervised visual pre-training. *arXiv:2011.09157*.
- Zhu, J., Xia, Y., Wu, L., Deng, J., Zhou, W., Qin, T., and Li, H. (2020). Masked contrastive representation learning for reinforcement learning. *arXiv:2010.07470*.

The appendix is organized as follows: [Appendix A-C](#) present proofs of our theoretical results. [Appendix D](#) presents experimental details including additional experiments and hyperparameter values.

A USEFUL FACTS

The standard multi-class hinge loss $\ell(v) = \max\{0, 1 + \max_i\{-v\}\}$ and the logistic loss $\ell(v) = \log(1 + \sum_i \exp(-v_i))$, for vector $v \in \mathbb{R}^k$, satisfy the following lemma:

Lemma 1 (Sub-additivity of losses). *Let $v \in \mathbb{R}^k$ be any vector. For all $I_1, I_2 \subset [k]$, with $S = I_1 \cup I_2$, we have that*

$$\ell(\{v_i\}_{i \in I_1}) \leq \ell(\{v_i\}_{i \in S}) \leq \ell(\{v_i\}_{i \in I_1}) + \ell(\{v_i\}_{i \in I_2}).$$

Lemma 2 (Coupon Collector). *Consider the coupon collector problem with n coupons, each with probability p_i . In each trial we collect a coupon by sampling from the distribution (p_1, \dots, p_n) . Let M denote the minimum number of trials until all coupons have been collected. Then:*

$$\mathbb{E}[M] = \int_0^\infty \left(1 - \prod_{i=1}^n (1 - \exp(-p_i t))\right) dt \leq \frac{H_n}{\min_i p_i},$$

where H_n is the n -th harmonic number. Furthermore, using Markov's inequality,

$$\Pr \left[M \geq \frac{2H_n}{\min_i p_i} \right] \leq \mathbb{E}[M] \cdot \frac{\min_i p_i}{2H_n} \leq 1/2.$$

Lemma 3. *For any vectors $v, w \in \mathbb{R}^t$,*

$$\ell(v) - \ell(w) \leq \left| \max_i (w_i - v_i) \right|.$$

Proof. For hinge loss,

$$\begin{aligned} \ell(v) &= \max \left(0, 1 + \max_i ((w_i - v_i) - w_i) \right) \\ &\leq \max \left(0, 1 + \max_i (-w_i) + \max_i (w_i - v_i) \right) \\ &\leq \max \left(0, 1 + \max_i (-w_i) \right) + \left| \max_i (w_i - v_i) \right| = \ell(w) + \left| \max_i (w_i - v_i) \right|. \end{aligned}$$

For logistic loss,

$$\begin{aligned} \ell(v) &= \log \left(1 + \sum_i \exp(-v_i) \right) \\ &= \log \left(1 + \sum_i \exp(-w_i) \exp(w_i - v_i) \right) \\ &\leq \log \left(1 + \exp \left(\left| \max_i (w_i - v_i) \right| \right) \sum_i \exp(-w_i) \right) \\ &\leq \left| \max_i (w_i - v_i) \right| + \log \left(1 + \sum_i \exp(-w_i) \right) = \ell(w) + \left| \max_i (w_i - v_i) \right|. \end{aligned}$$

The last inequality holds because $1 \leq \exp(\left| \max_i (w_i - v_i) \right|)$. □

B General Results with Non-uniform Probabilities and Proofs

In this section, we state and prove a more general version of [Theorem 1](#) and [Theorem 3](#) which can accommodate a non-uniform class distribution ρ . The particular results in the paper can be obtained by setting all class probabilities as $1/N$.

Notation. For a drawn sample, denote the collisions by $I(c, c_1^-, \dots, c_k^-) = \{i \in [k] | c = c_i^-\}$. Let $\tau_k(c) = \Pr_{c_i^- \sim \rho^k} [I(c, c_1^-, \dots, c_k^-) \neq \emptyset] = 1 - (1 - \rho(c))^k$ denote the probability of seeing class c in the negative samples (i.e., the collision probability). Also, let $\tau_k = \sum_{c' \in \mathcal{C}} \rho(c') \tau_k(c') = 1 - \sum_{c' \in \mathcal{C}} \rho(c') (1 - \rho(c'))^k$. We will drop the arguments of I when it is clear from the context.

B.1 General Transfer Theorem

Let us first start with the general transfer theorem.

Theorem 5 (General Transfer Bound). *For all $f \in \mathcal{F}$, we have*

$$\mathcal{L}^{\text{sup}}(f) \leq \frac{2 \max\left(1, \frac{2(1-\rho_{\min})H_{N-1}}{k\rho_{\min}}\right)}{(1-\rho_{\max})^k} \left(\mathcal{L}_{\text{NCE}}^{(k)}(f) - \tau_k \mathbb{E}_{c, c_i^- \sim \rho^{k+1}} [\ell_{|I|}(0) | I \neq \emptyset] \right),$$

where $\rho_{\min} = \min_c \rho(c)$, $\rho_{\max} = \max_c \rho(c)$ and H_t is the t^{th} harmonic number.

Proof. The first step is to apply Jensen's inequality to get,

$$\begin{aligned} \mathcal{L}_{\text{NCE}}^{(k)}(f) &= \mathbb{E}_{c, c_i^- \sim \rho^{k+1}} \mathbb{E}_{x, x^+ \sim D_c^2} \mathbb{E}_{x_i^- \sim D_{c_i^-}} [\ell(\{f(x)^\top (f(x^+) - f(x_i^-))\})] \\ &\geq \mathbb{E}_{c, c_i^- \sim \rho^{k+1}} \mathbb{E}_{x \sim D_c} [\ell(\{f(x)^\top (\mu_c - \mu_{c_i^-})\})]. \end{aligned}$$

Introducing the collision function I , we have

$$\begin{aligned} \mathcal{L}_{\text{NCE}}^{(k)}(f) &\geq \mathbb{E}_{x \sim D_c} \mathbb{E}_{c_i^- \sim \rho^k} [\ell(\{f(x)^\top (\mu_c - \mu_{c_i^-})\})] \\ &\geq \mathbb{E}_{(x, c) \sim D} \left[(1 - \tau_k(c)) \mathbb{E}_{c_i^- \sim \rho^k} [\ell(\{f(x)^\top (\mu_c - \mu_{c_i^-})\}) | I = \phi, c] \right] + \tau_k \mathbb{E}_{c, c_i^- \sim \rho^{k+1}} [\ell_{|I|}(0) | I \neq \emptyset] \end{aligned}$$

Here $\ell_t(0)$ is equal to ℓ evaluated at a t -dimensional 0 vector, which we obtain $\ell_{|I|}(0)$ by using the sub-additivity property in [Lemma 1](#). Note that for the second term, conditional on I , the distribution of c is not ρ in general. We deal with the second term in the next section, and focus now on the first term.

Lemma 4. *For any $c \in \mathcal{C}$ and x , we have,*

$$\mathbb{E}_{c_i^- \sim \rho^k} [\ell(\{f(x)^\top (\mu_c - \mu_{c_i^-})\}) | I = \emptyset, c] \geq \frac{1}{2 \left\lceil \frac{2(1-\rho(c))H_{N-1}}{k \min_{c' \neq c} \rho(c')} \right\rceil} \cdot \ell(\{f(x)^\top (\mu_c - \mu_{c'})\}_{c' \in \mathcal{C} \setminus c}),$$

where H_N is the N^{th} Harmonic number.

Proof. We can view the conditioning on $I = \emptyset$ and c as selecting c_i^- from ρ_{-c} with support on all $c' \neq c$ and $\rho_{-c}(c') = \frac{\rho(c')}{1-\rho(c)}$. Therefore, we have

$$\begin{aligned} &\mathbb{E}_{c_i^- \sim \rho^k} [\ell(\{f(x)^\top (\mu_c - \mu_{c_i^-})\}) | I = \emptyset, c] \\ &\stackrel{(a)}{=} \mathbb{E}_{c_i^- \sim \rho_{-c}^k} [\ell(\{f(x)^\top (\mu_c - \mu_{c_i^-})\})] \\ &\stackrel{(b)}{=} \frac{1}{m} \sum_{j=1}^m \mathbb{E}_{c_{ij}^- \sim \rho_{-c}^k} [\ell(\{f(x)^\top (\mu_c - \mu_{c_{ij}^-})\})] \\ &= \frac{1}{m} \mathbb{E}_{c_{ij}^- \sim \rho_{-c}^{k \times m}} \left[\sum_{j=1}^m \ell(\{f(x)^\top (\mu_c - \mu_{c_{ij}^-})\}) \right] \end{aligned}$$

$$\begin{aligned}
 &\stackrel{(c)}{\geq} \frac{1}{m} \mathbb{E}_{c_{ij}^- \sim \rho_{-c}^{k \times m}} \left[\ell \left(\{f(x)^\top (\mu_c - \mu_{c'})\}_{c' \in \cup \{c_{ij}^-\}} \right) \right] \\
 &\stackrel{(d)}{\geq} \frac{1}{m} \mathbb{E}_{c_{ij}^- \sim \rho_{-c}^{k \times m}} \left[\mathbf{1} [\cup \{c_{ij}^-\} = \mathcal{C} \setminus \{c\}] \ell \left(\{f(x)^\top (\mu_c - \mu_{c'})\}_{c' \in \mathcal{C} \setminus \{c\}} \right) \right] \\
 &= \frac{1}{m} \cdot \ell \left(\{f(x)^\top (\mu_c - \mu_{c'})\}_{c' \in \mathcal{C} \setminus \{c\}} \right) \cdot \Pr_{c_{ij}^- \sim \rho_{-c}^{k \times m}} [\cup \{c_{ij}^-\} = \mathcal{C} \setminus \{c\}].
 \end{aligned}$$

We replace the distribution over negatives, as described above to get (a). Then we construct m copies of this expression and take an average to get (b), which is a key step in the argument. Next, (c) uses [Lemma 1](#), which allows us to replace the sum over the m losses with a single multi-class loss on the union of the labels. Inequality (d) uses the non-negativity of the loss and finally we observe that the loss expression no longer depends on the samples c_{ij}^- .

To bound the probability term, note that it can be viewed as an instance of the standard coupon collector problem with $N - 1$ different coupons with non-identical probabilities of selection given by ρ_{-c} . In this instance, we perform mk trials. Therefore using [Lemma 2](#), if we take $mk \geq 2 \frac{(1-\rho(c))H_{N-1}}{\min_{c' \neq c} \rho(c')}$ boxes, we should cover all coupons with probability at least $1/2$. This implies taking $m = \left\lceil \frac{2(1-\rho(c))H_{N-1}}{k \min_{c' \neq c} \rho(c')} \right\rceil$, we get that the probability of hitting all elements in $\mathcal{C} \setminus \{c\}$ is at least $1/2$, which yields the desired result. \square

Substituting back, for the first term, we get,

$$\begin{aligned}
 &\mathbb{E}_{(x,c) \sim D} \left[(1 - \tau_k(c)) \mathbb{E}_{c_i^- \sim \rho^k} \left[\ell \left(\{f(x)^\top (\mu_c - \mu_{c_i^-})\} \right) \middle| I = \phi, c \right] \right] \\
 &\geq \mathbb{E}_{(x,c) \sim D} \left[\frac{1 - \tau_k(c)}{2 \left\lceil \frac{2(1-\rho(c))H_{N-1}}{k \min_{c' \neq c} \rho(c')} \right\rceil} \cdot \ell \left(\{f(x)^\top (\mu_c - \mu_{c'})\}_{c' \in \mathcal{C} \setminus \{c\}} \right) \right] \\
 &\geq \frac{(1 - \rho_{\max})^k}{2 \left\lceil \frac{2(1-\rho_{\min})H_{N-1}}{k \rho_{\min}} \right\rceil} \mathbb{E}_{(x,c) \sim D} \left[\ell \left(\{f(x)^\top (\mu_c - \mu_{c'})\}_{c' \in \mathcal{C} \setminus \{c\}} \right) \right] \\
 &\geq \frac{(1 - \rho_{\max})^k}{2 \left\lceil \frac{2(1-\rho_{\min})H_{N-1}}{k \rho_{\min}} \right\rceil} \mathcal{L}_{\text{sup}}^\mu(f).
 \end{aligned}$$

This gives us the desired result. \square

B.2 Refining the Transfer Theorem

As [Saunshi et al. \(2019\)](#) point out, we cannot drive \mathcal{L}_{NCE} to 0 as it is always lower bounded by τ_k . To get a better understanding of when the loss is actually small, let us define two terms that will be useful for a more refined analysis.

Definition 2 (Average Intra-class Variance). *We define the average intra-class variance of a learned embedding as,*

$$s(f) := \mathbb{E}_{c \sim \rho} \left[\mathbb{E}_{x \sim D_c} [\|f(x)\|] \sqrt{\|\Sigma(f, c)\|_2} \right]$$

where $\Sigma(f, c) = \mathbb{E}_{x \sim D_c} [(f(x) - \mu_c)(f(x) - \mu_c)^\top]$ is the covariance matrix of the representation f restricted to class D_c .

Definition 3 (NCE loss without collisions). *We define a loss function that is obtained from the NCE loss by removing the colliding negatives,*

$$\mathcal{L}_{\neq}^{(k)}(f) := \mathbb{E}_{\substack{c, c_i^- \sim \rho^{k+1} \\ x, x^+ \sim D_c^2 \\ x_i^- \sim D_{c_i^-}}} \left[\ell \left(\{f(x)^\top (f(x^+) - f(x_i^-))\}_{i \notin I} \right) \right]$$

Observe that both $\mathcal{L}_{\neq}^{(k)}$ as well as $s(f)$ can be made very small for large enough \mathcal{F} . Now we are ready to present our refined bound.

Theorem 6 (General Refined Bound). *For all $f \in \mathcal{F}$, we have*

$$\mathcal{L}^{\text{sup}}(f) \leq \frac{2 \left\lceil \frac{2(1-\rho_{\min})H_{N-1}}{k\rho_{\min}} \right\rceil}{(1-\rho_{\max})^k} \left(\mathcal{L}_{\neq}^{(k)}(f) + \sqrt{k\rho_{\max}} \cdot s(f) \right).$$

Proof. Decomposing the NCE loss gives us,

$$\begin{aligned} & \mathcal{L}_{\text{NCE}}^{(k)}(f) - \tau_k \mathbb{E}_{c, c_i^- \sim \rho^{k+1}} [\ell_{|I|}(0) | I \neq \emptyset] \\ & \leq \mathbb{E}_{\substack{c, c_i^- \sim \rho^{k+1} \\ x, x^+ \sim D_c^2 \\ x_i^- \sim D_{c_i^-}}} [\ell(\{f(x)^\top (f(x^+) - f(x_i^-))\}_{i \notin I}) + \ell(\{f(x)^\top (f(x^+) - f(x_i^-))\}_{i \in I})] - \tau_k \mathbb{E}_{c, c_i^- \sim \rho^{k+1}} [\ell_{|I|}(0) | I \neq \emptyset] \\ & = \mathbb{E}_{\substack{c, c_i^- \sim \rho^{k+1} \\ x, x^+ \sim D_c^2 \\ x_i^- \sim D_{c_i^-}}} [\ell(\{f(x)^\top (f(x^+) - f(x_i^-))\}_{i \notin I})] + \tau_k \underbrace{\mathbb{E}_{\substack{c, c_i^- \sim \rho^{k+1} \\ x, x^+ \sim D_c^2 \\ x_i^- \sim D_{c_i^-}}} [\ell(\{f(x)^\top (f(x^+) - f(x_i^-))\}_{i \in I}) - \ell_{|I|}(0) | I \neq \emptyset]}_{\Delta(f)} \\ & = \mathcal{L}_{\neq}^{(k)} + \tau_k \Delta(f). \end{aligned}$$

As before, note that the second term samples the classes (c, c_i^-) from the conditional distribution given that there is a collision. We follow the convention that an empty set implies value 0 for $\mathcal{L}_{\neq}^{(k)}$. Note that this term can be made very small if the classes are separable by f with a sufficient margin. As for the second term, we will first prove the following lemma,

Lemma 5. *For any $t \geq 0$:*

$$\mathbb{E}_{x, x^+, x_i^- \sim D_c^{2+t}} [\ell(\{f(x)^\top (f(x^+) - f(x_i^-))\}_{i \in [t]}) - \ell_t(0)] \leq \sqrt{t} \mathbb{E}_{x \sim D_c} [\|f(x)\|^2] \sqrt{\|\Sigma(f, c)\|_2}. \quad (1)$$

The above lemma is an improvement to Lemma A.1 of [Saunshi et al. \(2019\)](#) which has a linear scaling with t . Observe that on the LHS, all samples are drawn from D_c , which corresponds to the situation where we have t collisions with the anchor class c .

Proof of Lemma 5. From [Lemma 3](#), we know that

$$\mathbb{E}_{x, x^+, x_i^- \sim D_c^{2+t}} [\ell(\{f(x)^\top (f(x^+) - f(x_i^-))\}_{i \in [t]}) - \ell_t(0)] \leq \mathbb{E}_{x, x^+, x_i^- \sim D_c^{2+t}} \left[\left| \max_i f(x)^\top (f(x_i^-) - f(x^+)) \right| \right].$$

Now we have,

$$\begin{aligned} & \mathbb{E}_{x, x^+, x_i^- \sim D_c^{2+t}} \left[\left| \max_i f(x)^\top (f(x_i^-) - f(x^+)) \right| \right] \\ & = \mathbb{E}_{x \sim D_c} \left[\|f(x)\| \mathbb{E}_{x^+, x_i^- \sim D_c^{1+t}} \left[\left| \max_i \frac{f(x)^\top}{\|f(x)\|} (f(x_i^-) - f(x^+)) \right| \right] \right] \\ & \leq \mathbb{E}_{x \sim D_c} \left[\|f(x)\| \sqrt{\mathbb{E}_{x^+, x_i^- \sim D_c^{1+t}} \left[\left(\max_i \frac{f(x)^\top}{\|f(x)\|} (f(x_i^-) - f(x^+)) \right)^2 \right]} \right] \\ & \leq \mathbb{E}_{x \sim D_c} \left[\|f(x)\| \sqrt{\sum_i \mathbb{E}_{x^+, x_i^- \sim D_c^2} \left[\left(\frac{f(x)^\top}{\|f(x)\|} (f(x_i^-) - f(x^+)) \right)^2 \right]} \right] \end{aligned}$$

$$\begin{aligned}
 &\leq \mathbb{E}_{x \sim D_c} \left[\|f(x)\| \sqrt{t \mathbb{E}_{x^+, x^- \sim D_c^2} \left[\left(\frac{f(x)^\top}{\|f(x)\|} (f(x^-) - f(x^+)) \right)^2 \right]} \right] \\
 &\leq \sqrt{t} \mathbb{E}_{x \sim D_c} [\|f(x)\|] \sqrt{\|\Sigma(f, c)\|_2}.
 \end{aligned}$$

Combining with the above inequality gives us the desired result. \square

Substituting back, we get the following bound on the second term,

$$\begin{aligned}
 \tau_k \Delta(f) &\leq \mathbb{E}_{c \sim \rho} \left[\tau_k(c) \mathbb{E}_{x \sim D_c} [\|f(x)\|] \sqrt{\|\Sigma(f, c)\|_2} \mathbb{E}_{c_i^- \sim \rho^k} \left[\sqrt{|I|} \mathbb{1}[I \neq \emptyset, c] \right] \right] \\
 &\leq \mathbb{E}_{c \sim \rho} \left[\tau_k(c) \mathbb{E}_{x \sim D_c} [\|f(x)\|] \sqrt{\|\Sigma(f, c)\|_2} \sqrt{\mathbb{E}_{c_i^- \sim \rho^k} [|I| \mathbb{1}[I \neq \emptyset, c]]} \right] \\
 &\stackrel{(a)}{=} \mathbb{E}_{c \sim \rho} \left[\mathbb{E}_{x \sim D_c} [\|f(x)\|] \sqrt{\|\Sigma(f, c)\|_2} \sqrt{k\rho(c)\tau_k(c)} \right] \\
 &= \sqrt{k} \mathbb{E}_{c \sim \rho} \left[\mathbb{E}_{x \sim D_c} [\|f(x)\|] \sqrt{\|\Sigma(f, c)\|_2} \sqrt{\rho(c)\tau_k(c)} \right] \\
 &\stackrel{(b)}{\leq} \sqrt{k\rho_{\max} s(f)}
 \end{aligned}$$

where (a) follows from observing that $\mathbb{E}_{c_i^- \sim \rho^k} [|I| \mathbb{1}[I \neq \emptyset, c]] = \frac{k\rho(c)}{\tau_k(c)}$ (see also equation (35) of Saunshi et al. (2019)) and (b) follows from Definition 2. Combining these terms gives us the desired result. \square

Note that Saunshi et al. (2019) also obtain a similar result, but in their analysis the coefficient on the $s(f)$ term scales linearly with k . Instead, we obtain a \sqrt{k} scaling. However, for the case of uniform class distribution ρ , their coefficient is tighter in the $k \leq n$ regime.

B.3 Generalization Bound

To transfer our guarantee to the minimizer of the empirical NCE loss, we prove a generalization bound using a notion of worst case Rademacher complexity.

Definition 4 (Rademacher Complexity). *The empirical Rademacher complexity of a real-valued function class \mathcal{H} for a set S drawn from \mathcal{X}^m is defined to be*

$$\widehat{\mathcal{R}}_S(\mathcal{H}) := \mathbb{E}_\sigma \left[\sup_{h \in \mathcal{H}} \left(\frac{1}{m} \sum_{i=1}^m \sigma_i h(x_i) \right) \right],$$

where σ_i are iid Rademacher random variables. We define the worst-case Rademacher complexity to be

$$\mathcal{R}_m(\mathcal{H}) := \max_{S \sim \mathcal{X}^m} \widehat{\mathcal{R}}_S(\mathcal{H}).$$

We will use the following theorem to bound the Rademacher complexity of our vector valued class,

Theorem 7 (Foster and Rakhlin (2019)). *Let $\mathcal{H} \subseteq \{h : \mathcal{X} \rightarrow \mathbb{R}^K\}$, and let ϕ be a real-valued L -Lipschitz function with respect to the ℓ_∞ norm. We have,*

$$\mathcal{R}_S(\phi \circ \mathcal{H}) \leq \tilde{O} \left(L\sqrt{k} \right) \max_i \mathcal{R}_M(\mathcal{H}|_i)$$

where $\mathcal{H}|_i$ is the function class restricted to the i^{th} coordinate.

Note that the above theorem relates the empirical Rademacher complexity on sample S to the worst case Rademacher complexity. Finally we will use the following standard generalization result based on Rademacher complexity,

Theorem 8 (Mohri et al. (2018)). For a real-valued function class \mathcal{H} and a set $S = \{x_1, \dots, x_m\} \in \mathcal{X}^m$ drawn i.i.d. from some distribution, with probability $1 - \delta$, for all $h \in \mathcal{H}$

$$\left| \mathbb{E}[h(x)] - \frac{1}{m} \sum_{i=1}^m h(x_i) \right| \leq \mathcal{R}_S(\mathcal{H}) + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right).$$

Now we state our generalization bound for the NCE loss.

Theorem 9. Let $B = \max_{x \in \mathcal{X}, f' \in \mathcal{F}} \|f'(x)\|_1$. For all $f \in \mathcal{F}$, with probability $1 - \delta$ over the random sample S of M NCE examples used to fit \hat{f} ,

$$\mathcal{L}_{\text{NCE}}^{(k)}(\hat{f}) \leq \mathcal{L}_{\text{NCE}}^{(k)}(f) + \tilde{O}\left(B\sqrt{(k+2)d}\right) \max_i \mathcal{R}_M(\mathcal{F}|_i) + O\left(\sqrt{\frac{\log(1/\delta)}{M}}\right).$$

Proof. Let us assume the dimension of input features is n . Define function $\phi : \mathbb{R}^{d(k+2)} \rightarrow \mathbb{R}$ as

$$\phi(z, z^+, z_1^-, \dots, z_k^-) = \ell\left(\{z^T(z^+ - z_i^-)\}_{i=1}^k\right)$$

where each $z, z^+, z_i^- \in \mathbb{R}^d$. Also define the class of functions $\mathcal{G} := \{g_f | f \in \mathcal{F}\}$ where $g_f : \mathbb{R}^{n(k+2)} \rightarrow \mathbb{R}^{d(k+2)}$ such that

$$g_f(x, x^+, x_1^-, \dots, x_k^-) = (f(x), f(x^+), f(x_1^-), \dots, f(x_k^-)).$$

Observe that $\phi \circ g_f$ gives us exactly the NCE loss for a sample. Therefore using [Theorem 8](#) and the standard ERM analysis, we have with probability $1 - \delta$ for all $f \in \mathcal{F}$,

$$\mathcal{L}_{\text{NCE}}^{(k)}(\hat{f}) \leq \mathcal{L}_{\text{NCE}}^{(k)}(f) + 2\mathcal{R}_S(\phi \circ \mathcal{G}) + O\left(\sqrt{\frac{\log(1/\delta)}{M}}\right).$$

Now we will show that ϕ is Lipschitz. Observe that, for any $W = (w, w^+, w_1^-, \dots, w_k^-), Z = (z, z^+, z_1^-, \dots, z_k^-)$,

$$\begin{aligned} |\phi(Z) - \phi(W)| &= |\phi(z, z^+, z_1^-, \dots, z_k^-) - \phi(w, w^+, w_1^-, \dots, w_k^-)| \\ &= \left| \ell\left(\{z^T(z^+ - z_i^-)\}_{i=1}^k\right) - \ell\left(\{w^T(w^+ - w_i^-)\}_{i=1}^k\right) \right| \\ &\leq \left| \max_i (w^T(w^+ - w_i^-) - z^T(z^+ - z_i^-)) \right| && \text{(Using Lemma 3)} \\ &\leq \left| w^T w^+ - z^T z^+ + \max_i z^T z_i^- - w^T w_i^- \right| \\ &\leq |w^T w^+ - z^T z^+| + \max_i |w^T w_i^- - z^T z_i^-| \end{aligned}$$

Thus we have several terms with a similar form $|w^T \tilde{w} - z^T \tilde{z}|$ where $\tilde{w} \in W, \tilde{z} \in Z$. For these, we bound as

$$|w^T \tilde{w} - z^T \tilde{z}| \leq |w^T(\tilde{w} - \tilde{z}) + (w - z)^T \tilde{z}| \leq 2B\|W - Z\|_\infty,$$

where $\|\cdot\|_\infty$ is the ℓ_∞ norm in the $d(k+2)$ dimensional space. Thus we conclude that

$$|\phi(Z) - \phi(W)| \leq 4B\|W - Z\|_\infty.$$

Now applying [Theorem 7](#), we have

$$\mathcal{R}_S(\phi \circ \mathcal{G}) \leq \tilde{O}\left(B\sqrt{(k+2)d}\right) \max_{i \in [d(k+2)]} \mathcal{R}_M(\mathcal{G}|_i) \leq \tilde{O}\left(B\sqrt{(k+2)d}\right) \max_{i \in [d]} \mathcal{R}_M(\mathcal{F}|_i).$$

Substituting back gives us the desired result. Here, to pass from the Rademacher complexity of \mathcal{G} , which has output dimension $d(k+2)$, we use the fact that g_f is simply $k+2$ copies of f applied to different examples and that we are working with the worst-case Rademacher complexity. \square

Applying this for \hat{f} and using the fact that it is the minimizer of the empirical NCE loss gives us the required guarantee on the NCE loss of \hat{f} .

C CALCULATIONS FOR THE EXAMPLE IN SECTION 4

Properties of f_1 . First, it is easy to see that $\mathcal{L}_{\text{sup}}^\mu(f_1) = 1$ for the hinge loss, since the mean embeddings satisfy $\mu_{2i} = \mu_{2i+1} = e_i$. In addition, every example is embedded as e_j for some j , which means that $W^\mu e_j$ is a 2-sparse binary vector. Such score vectors always yield hinge loss 1.

Now, let us verify the stated identity for $\mathcal{L}_{\text{NCE}}^{(k)}(f_1)$. Due to symmetry, we can assume that x, x^+ come from class 1. Now, using the definition of f_1 we see that

$$\max\{0, 1 + f_1(x)^\top (f_1(x_i^-) - f_1(x^+))\} = \mathbf{1}\{c_i^- \in \{1, 2\}\}.$$

Taking max over all of the negatives, we see that the NCE loss will be 1 if and only if class 1 or 2 appear in the negatives and otherwise it will be 0. Formally

$$\mathcal{L}_{\text{NCE}}^{(k)}(f_1) = \mathbb{P}_{S \sim \rho^k}[S \cap \{1, 2\} \neq \emptyset] = 1 - \mathbb{P}_{S \sim \rho^k}[S \cap \{1, 2\} = \emptyset] = 1 - (1 - 2/N)^k,$$

where in the last step we use that the prior ρ is uniform. Now the stated identity follows by algebraic manipulations. In particular:

$$\begin{aligned} \mathcal{L}_{\text{NCE}}^{(k)}(f_1) - \tau_k &= 1 - (1 - 2/N)^k - 1 + (1 - 1/N)^k = \left(\frac{N-1}{N}\right)^k \left(1 - \left(1 - \frac{1}{N-1}\right)^k\right) \\ &= (1 - \tau_k) \left(1 - \left(1 - \frac{1}{N-1}\right)^k\right) \end{aligned}$$

Finally, observe that

$$\lim_{k \rightarrow \infty} \mathcal{L}_{\text{NCE}}^{(k)}(f_1) = 1 - \lim_{k \rightarrow \infty} (1 - 2/N)^k = 1.$$

Note that for f_1 ,

$$\mathcal{L}_{\text{sup}}^\mu = 1 = \frac{1}{(1 - \tau_k) \left(1 - \left(1 - \frac{1}{N-1}\right)^k\right)} \left(\mathcal{L}_{\text{NCE}}^{(k)}(f_1) - \tau_k\right) \geq \frac{N-1}{(1 - \tau_k)k} \left(\mathcal{L}_{\text{NCE}}^{(k)}(f_1) - \tau_k\right)$$

where the inequality follows from observing $1 - (1 - x)^r \leq rx$ for $0 < x < 1$. This shows that [Theorem 1](#) is tight up to log factors.

Properties of f_2 . For f_2 , the supervised loss is computed as follows. First note that $\mu_i = e_i$ for each class. Now, observe that if the example x is from class i we have that $f_2(x)^\top \mu_j = 0$ for all $j \neq i$. For class i we have $f_2(x)^\top \mu_i \sim \text{Unif}(\{1 - \epsilon, 1 + \epsilon\})$ where the randomness is over the realization of x . In the first of these cases, the hinge loss is ϵ , while in the second case it is 0. Thus, we have $\mathcal{L}_{\text{sup}}^\mu(f_2) = \epsilon/2$.

To analyze \mathcal{L}_{NCE} we consider two cases. Again, due to symmetry we can assume that x, x^+ come from class 1.

Case 1, no collisions: If none of the negative examples come from class 1 then $f_2(x)^\top f_2(x'_i) = 0$ for all i , hence the hinge loss is (where σ, σ^+ are Rademacher random variables):

$$\mathbb{E}_{\sigma, \sigma^+} \max\{0, 1 + (1 + \sigma\epsilon)(-1 - \sigma^+\epsilon)\} = \epsilon^2/4 + \epsilon/2.$$

(With probability 3/4 over the Rademacher variables, the hinge structure clips the loss at 0.)

Case 2, collisions: We provide crude upper and lower bounds in the case of collisions. Assume that x_i^- is a collision, so that this term in the hinge loss is:

$$\begin{aligned} \mathbb{E}_{\sigma, \sigma^+, \sigma_i^-} \max\{0, 1 + (1 + \sigma\epsilon)(1 + \sigma_i^- \epsilon - 1 - \sigma^+\epsilon)\} &= \mathbb{E}_{\sigma, \sigma^+, \sigma_i^-} \max\{0, 1 + (1 + \sigma\epsilon)((\sigma_i^- - \sigma^+)\epsilon)\} \\ &\geq \max\{0, \mathbb{E}_{\sigma, \sigma^+, \sigma_i^-} 1 + (1 + \sigma\epsilon)((\sigma_i^- - \sigma^+)\epsilon)\} = 1. \end{aligned}$$

Here, the inequality is Jensen's inequality and then we use independence of the Rademacher random variables. This is clearly a lower bound on the expected NCE loss when there is a collision. We can also upper bound the loss when there is a collision by setting $\sigma_i^+ = 1$

$$\mathbb{E}_{\sigma, \sigma^+, \sigma_i^-} \max\{0, 1 + (1 + \sigma\epsilon)((\sigma_i^- - \sigma^+)\epsilon)\} \leq \mathbb{E}_{\sigma, \sigma^+} \max\{0, 1 + (1 + \sigma\epsilon)(1 - \sigma^+)\epsilon\} = 1 + \epsilon.$$

In addition, since $\epsilon \leq 1$, for any values of σ, σ^+ , this term dominates the loss incurred from any other negative. Thus this is an upper bound on the NCE-loss when there is a collision. Thus, we have the bounds

$$(1 - \tau_k)(\epsilon^2/4 + \epsilon/2) + \tau_k \leq \mathcal{L}_{\text{NCE}}^{(k)}(f_2) \leq (1 - \tau_k)(\epsilon^2/4 + \epsilon/2) + \tau_k(1 + \epsilon).$$

Actually we can obtain a much better lower bound. Note that the smallest realization of a collision loss is $1 + (1 + \epsilon)(-2\epsilon) = 1 - 2\epsilon - 2\epsilon^2$. This quadratic is positive for $\epsilon \in [0, (\sqrt{3} - 1)/2]$, which we assume going forward. With this choice of ϵ , there will be no clipping and we can ignore taking the max with 0. Next, observe that the loss we incur for a non-colliding example is $\max\{0, 1 + (1 + \sigma\epsilon)(-1 - \sigma^+\epsilon)\}$. We can see that for any realizations of $\sigma, \sigma^+, \sigma_i^-$

$$1 + (1 + \sigma\epsilon)(-1 - \sigma^+\epsilon) \leq 1 + (1 + \sigma\epsilon)(\sigma_i^- - \sigma^+)\epsilon,$$

which means that we can ignore all of the non-colliding examples when evaluating the loss. Thus, the loss is

$$\mathbb{E} \max\{0, 1 + (1 + \sigma\epsilon)(\max_{i \in I} \sigma_i^- - \sigma^+)\epsilon\} = 1 + \epsilon \cdot \mathbb{E} \max_{i \in I} \sigma_i^- ,$$

where I is the set of colliding examples. This is clearly upper bounded by $\tau_k(1 + \epsilon)$ after taking into account the probability of a collision. For a lower bound, we have

$$\begin{aligned} \mathbb{E} \left[\mathbf{1}\{|I| > 0\} (1 + \epsilon \max_{i \in I} \sigma_i) \right] &= \tau_k + \epsilon \mathbb{E} \left[\mathbf{1}\{|I| > 0\} \max_{i \in I} \sigma_i \right] \\ &= \tau_k + \epsilon \left(\sum_{t=1}^k \binom{k}{t} \left(\frac{1}{N}\right)^t \left(1 - \frac{1}{N}\right)^{k-t} \left((1 - (1/2)^t) \cdot 1 + (1/2)^t \cdot (-1) \right) \right) \\ &= \tau_k + \epsilon \left(\sum_{t=1}^k \binom{k}{t} \left(\frac{1}{N}\right)^t \left(1 - \frac{1}{N}\right)^{k-t} (1 - 2(1/2)^t) \right) \\ &= \tau_k + \epsilon \left(\tau_k - 2 \sum_{t=1}^k \binom{k}{t} \left(\frac{1}{N}\right)^t \left(1 - \frac{1}{N}\right)^{k-t} (1/2)^t \right) \\ &\geq \tau_k + \epsilon \left(\tau_k - \frac{k}{N} \left(1 - \frac{1}{N}\right)^{k-1} - \frac{1}{2} \sum_{t=2}^k \binom{k}{t} \left(\frac{1}{N}\right)^t \left(1 - \frac{1}{N}\right)^{k-t} \right) \\ &= \tau_k + \epsilon/2 \cdot \left(\tau_k - \frac{k}{N} \left(1 - \frac{1}{N}\right)^{k-1} \right). \end{aligned}$$

Here the only lower bound uses that in the terms where $t \geq 2$ we can upper bound $(1/2)^t \leq 1/4$. Combining this with the term from the ‘‘no-collision’’ case, we have

$$\begin{aligned} \mathcal{L}_{\text{NCE}}^{(k)}(f_2) &\geq (1 - \tau_k)(\epsilon^2/4 + \epsilon/2) + \tau_k + \epsilon/2 \left(\tau_k - \frac{k}{n} \left(1 - \frac{1}{N}\right)^{k-1} \right) \\ &= \tau_k + (1 - \tau_k)\epsilon^2/4 + \epsilon/2 \left(1 - \frac{k}{N} \left(1 - \frac{1}{N}\right)^{k-1} \right). \end{aligned}$$

Finally, notice that the upper bound on the collision case is tight if there is a collision that has $\sigma_i^+ = 1$. In the asymptotic regime where $k \rightarrow \infty$ we have that the probability of such a collision goes to 1, and simultaneously $\tau_k \rightarrow 1$. Thus in this limit this upper bound is tight, so we obtain $\lim_{k \rightarrow \infty} \mathcal{L}_{\text{NCE}}^{(k)}(f_2) = 1 + \epsilon$.

Dependence on k/N . Next, we claim that (bounds on) the first crossing point between the NCE losses for f_1 and f_2 depend on k only through the function k/N . To see why, recall the elementary inequalities

$$1 - x \leq (1 - x/n)^n \leq \exp(-x) \leq 1 - x + x^2$$

This means that a sufficient condition for f_1 to be selected by NCE is

$$\frac{2k}{N} \leq (1 - k/N)(\epsilon^2/4 + \epsilon/2) + (k/N - k^2/N^2)$$

While a sufficient condition for f_2 to be selected is

$$2k/N - 4k^2/N^2 \geq (1 - k/N + k^2/N^2)(\epsilon^2/4 + \epsilon/2) + k/N(1 + \epsilon)$$

These are both quadratic inequalities in k/N . Thus if they admit solutions, these would have k scaling linearly with N .

D EXPERIMENT DETAILS

D.1 Wiki-3029 NLP Experiments

We present additional details for the Wiki-3029 results below.

Dataset details. We use the dataset provided by Saunshi et al. (2019).⁴ This dataset was released by the authors, and we also explicitly obtained their permission to use the dataset for this study. We use the scripts provided to us by the authors to generate the Wiki-3029 dataset.

Training details. We perform NCE training for a fixed number of epochs. In each epoch, we take sentences from each article and pair them randomly to form a set of positively aligned examples (x, x^+) . We mix pairs across classes and divide this dataset into batches of a given size B and fit the representation using the Adam optimizer on the NCE loss. Formally, for a given batch $\mathcal{B} = \{(x_i, x_i^+)\}_{i=1}^B$ we optimize the loss:

$$\mathcal{L}_{\text{nce}}(\theta) = -\frac{1}{2B} \sum_{i=1}^B \ln \frac{w(x_i, x_i^+)}{\sum_{j \neq i} \{w(x_i, x_j^+) + w(x_i, x_j)\}} - \frac{1}{2B} \sum_{i=1}^B \ln \frac{w(x_i, x_i^+)}{\sum_{j \neq i} \{w(x_i^+, x_j^+) + w(x_i^+, x_j)\}},$$

where $w(x, x') = \exp(f_\theta(x)^\top f_\theta(x')/\tau)$, θ is the parameters of the bag-of-words model f , and τ is a temperature hyperparameter. The parameters θ consists of word embedding for every token in the vocabulary including the UNK token.

At the end of NCE training, we compute a mean classifier vector v_c for an article c by averaging $f_\theta(x)$ for every sentence x from article c in the training dataset. For a given sentence x' in the test dataset, we assign it the class $\arg \max_c f_\theta(x')^\top v_c$.

We further train a linear classifier on top of $f_\theta(x)$ for a fixed number of epochs. We optimize the logistic loss and use Adam optimizer. Formally, we use the entire NCE training dataset and do mini-batch learning. Given a batch $\{(x_i, c_i)\}_{i=1}^B$ of pairs i and sentences c_i we optimize the loss:

$$\mathcal{L}_{\text{downstream}}(W) = -\frac{1}{B} \sum_{i=1}^B \ln \frac{\exp((W^\top f_\theta(x_i))_{c_i})}{\sum_{c'} \exp((W^\top f_\theta(x_i))_{c'})},$$

where W is the parameters of the linear classifier. We keep the parameters θ fixed when training the linear classifier. We save the linear classifier at the end of each epoch and compute the aggregate training loss recorded in this epoch. Finally, we report performance of the model with the smallest aggregate training loss.

We initialize W and θ randomly using default PyTorch initialization.

Hyperparameters. We used the following hyperparameters for the Wiki-3029 experiments.

⁴<https://nlp.cs.princeton.edu/CURL/raw/>

Hyperparameters	Values
Word Embedding	768
Optimization method	Adam
Parameter initialization	PyTorch 1.4 default
Temperature	1
Gradient clip norm	2.5
NCE epochs	50
Downstream training epoch	50
Learning rate	0.01
Model architecture	Bag of Words
Data augmentation	None

Table 1: Hyperparameters for Wiki-3029

Detailed Figures with Error Bars. We show a detailed version of the left plot in Figure 3 in Figure 5. This shows the mean classifier accuracy on varying N and k for a fixed value of $B = 1000$. The gap between the best and worst performance in each row is quite noticeable (≈ 2 -3%) and this is much larger than the standard deviation (≤ 0.25). We also plot the performance of the trained linear classifier in Figure 6. Note that the general trend seems to be similar to Figure 5 and an intermediate value of k seems to be best for any value of N . However, we see that in this case the value of $k = 50$ is optimal across all values of N .

We show a detailed version of the center plot in Figure 3 in Figure 7. This shows the mean and trained classifier accuracy for different values of B, k , and N . We observe that gap between the best and worst performance is noticeable ($\approx 1\%$ for the mean classifier and ≈ 2 -3% for the trained classifier). Further, standard deviation is sufficiently low to enable reading trends from the mean performance. For the trained classifier, we observe that optimal performance is received with largest value of B and an intermediate value of k similar to the mean classifier.

Compute Infrastructure. We run our experiments on a cluster containing a mixture of P40, P100, and V100 GPUs. Each experiment runs on a single GPU in a docker container and takes less than 5 hours to complete with a V100 GPU.

D.2 Vision Experiment Details

Vision experiments used the same infrastructure as NLP experiments, but took roughly two days to fully run. We train models on the NCE task for 400 epochs. We used the following hyperparameters to run the included CIFAR-10 experiments:

Hyperparameters	Values
Latent dimension	128
Gradient clip norm	None
Optimization method	Adam
Parameter initialization	PyTorch 1.4 default
Learning rate	10^{-4}
Model architecture	ResNet-18

Table 2: Hyperparameters for vision

Additional Experiments. Here we provide a companion plot to Figure 4, but showing raw accuracies rather than normalized percentages (Figure 8). We then present corresponding results using a mean classifier rather than a trained, linear classifier for the downstream task. Figure 9 and Figure 10 respectively show values as a percent deviation from the highest-performing k for a given batch and raw accuracies.

The trends in these plots is similar to Figure 4, with results varying based on both dataset and whether or not

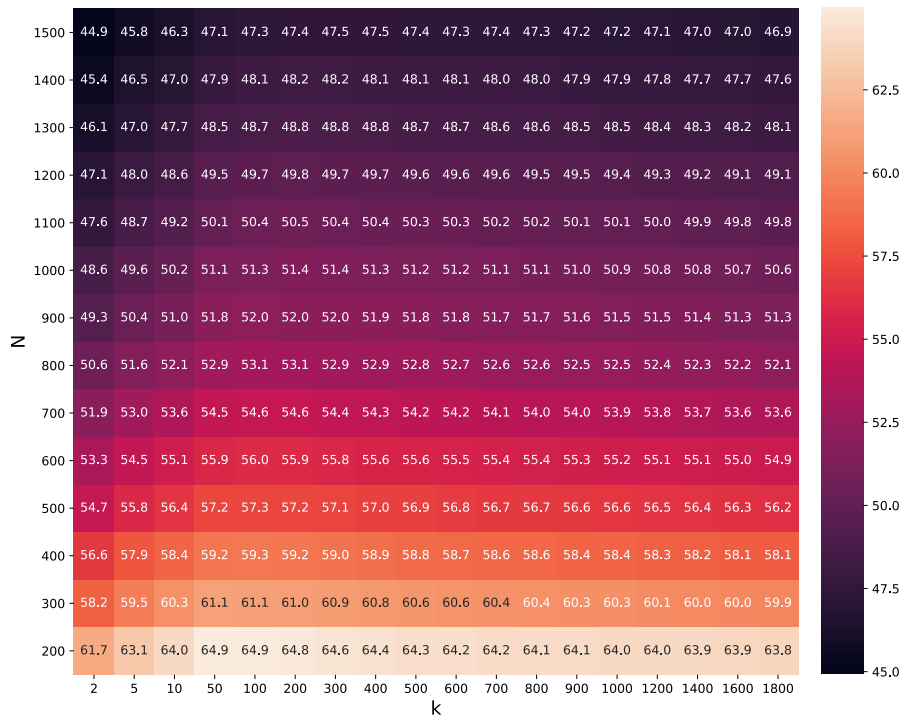


Figure 5: Mean classifier accuracy on the Wiki-3029 dataset for different values of number of classes (N) and number of negative examples (k), and a fixed batch size of 1000. Values show average performance across 5 trials with different seeds, and all standard deviations are at most 0.25 and typically much smaller.

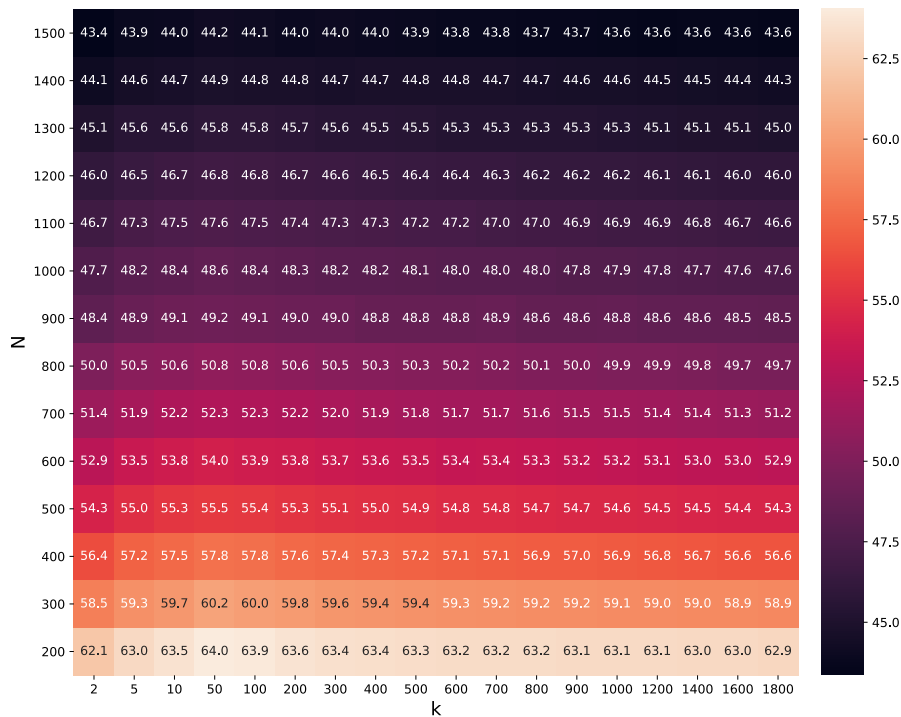


Figure 6: Trained linear classifier accuracy on the Wiki-3029 dataset for different values of number of classes (N) and number of negative examples (k), and a fixed batch size of 1000. Values show average performance across 5 trials with different seeds and all standard deviations are at most 0.28 and typically much smaller.

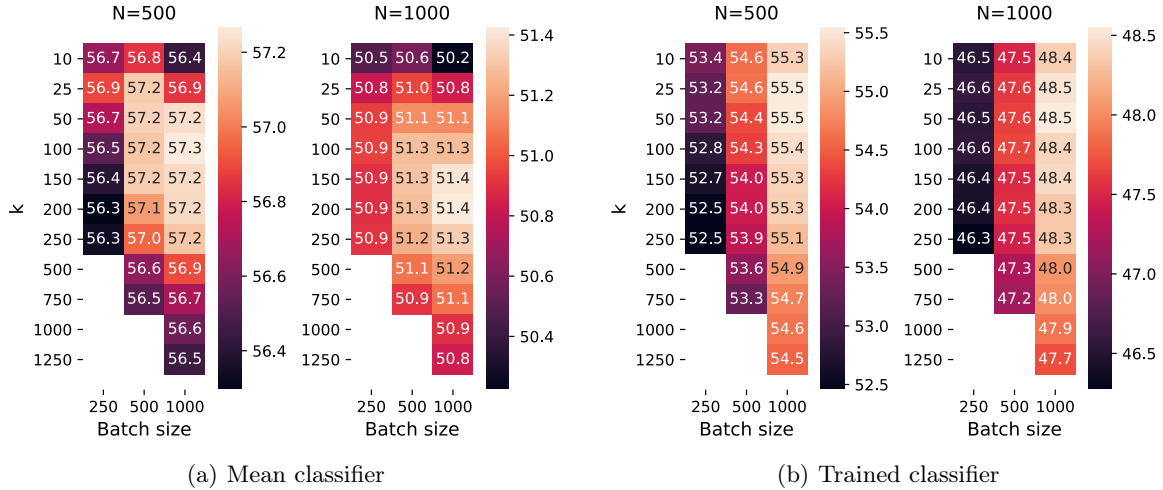


Figure 7: Mean and trained classifier accuracy on Wiki-3029 dataset for different values of batch size B and number of negatives examples k and two values of the number of classes N . **Left:** mean classifier performance (all standard deviations are less than 0.16). **Right:** linear classifier performance (all standard deviations are less than 0.26).

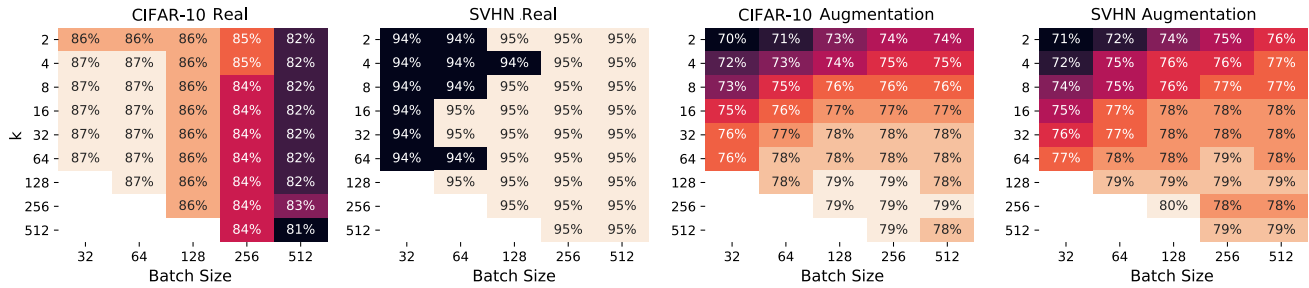


Figure 8: A companion to Figure 4, showing raw accuracies. **Left:** CIFAR-10 data, using a true positive sample from the same class, as used in our analysis. **Center Left:** SVHN data, also using a true positive. The rightmost figures are CIFAR-10 and SVHN respectively, where the positive sample is an augmented version of the reference sample, and where negatives are augmented as well.

augmentation is present. Interestingly, mean classifier performance on SVHN with augmented data is significantly worse than the trained classifier version in Figure 4. This is possibly due to the fact that the augmentation was not designed for digit data, including, for example, horizontal flips.

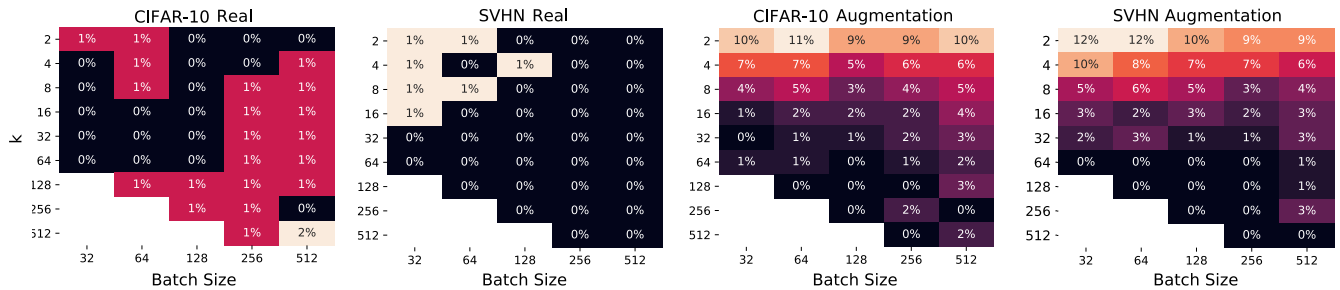


Figure 9: Average mean classifier accuracy as a function of both batch size and k for various vision datasets and NCE parameters. **Left:** CIFAR-10 data, using a true positive sample from the same class, as used in our analysis. **Center Left:** SVHN data, also using a true positive. The rightmost figures are CIFAR-10 and SVHN respectively, where the positive sample is an augmented version of the reference sample, and where negatives are augmented as well. Accuracies are represented as a percent difference from the highest accuracy in the column.

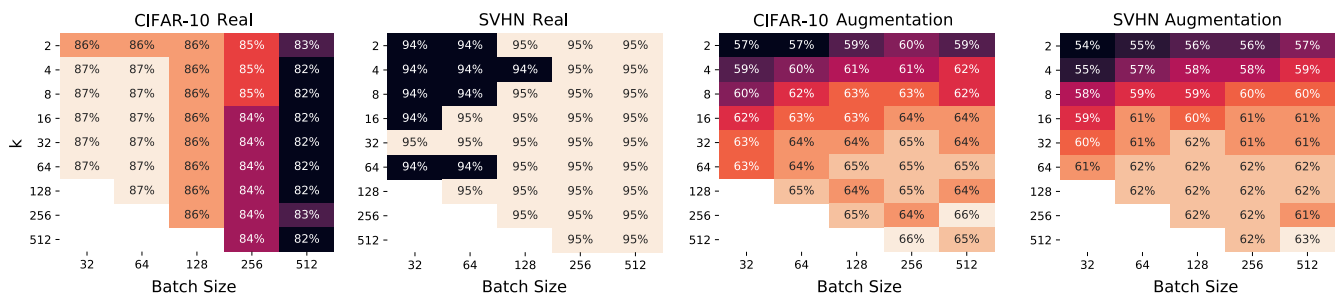


Figure 10: A companion to Figure 9, showing raw accuracies for the mean classifier. **Left:** CIFAR-10 data, using a true positive sample from the same class, as used in our analysis. **Center Left:** SVHN data, also using a true positive. The rightmost figures are CIFAR-10 and SVHN respectively, where the positive sample is an augmented version of the reference sample, and where negatives are augmented as well.