Noise contrastive estimation with soft targets for conditional models

Johannes Hugger
European Bioinformatics Institute
European Molecular Biology Laboratory
jhugger@ebi.ac.uk

Virginie Uhlmann
European Bioinformatics Institute
European Molecular Biology Laboratory
uhlmann@ebi.ac.uk

Abstract

Soft targets combined with the cross-entropy loss have shown to improve generalization performance of deep neural networks on supervised classification tasks. The standard cross-entropy loss however assumes data to be categorically distributed, which may often not be the case in practice. In contrast, InfoNCE does not rely on such an explicit assumption but instead implicitly estimates the true conditional through negative sampling. Unfortunately, it cannot be combined with soft targets in its standard formulation, hindering its use in combination with sophisticated training strategies. In this paper, we address this limitation by proposing a principled loss function that is compatible with probabilistic targets. Our new soft target InfoNCE loss is conceptually simple, efficient to compute, and can be derived within the framework of noise contrastive estimation. Using a toy example, we demonstrate shortcomings of the categorical distribution assumption of cross-entropy, and discuss implications of sampling from soft distributions. We observe that soft target InfoNCE performs on par with strong soft target cross-entropy baselines and outperforms hard target NLL and InfoNCE losses on popular benchmarks, including ImageNet. Finally, we provide a simple implementation of our loss, geared towards supervised classification and fully compatible with deep classification model trained with cross-entropy.

1 Introduction

The cross-entropy loss, or log loss, and its soft target variants are amongst the most popular objective functions for conditional density estimation in supervised classification problems He et al. [2016], Dosovitskiy et al. [2020]. In its base variant, it assumes a degenerate one-hot distribution $p(l|x) = \delta(l, k)$ as the underlying conditional density over labels l given data x, which might not suffice to capture the nature of complex or ambiguously annotated datasets. This simplifying assumption can have a negative effect on parameter estimation as well as model calibration properties Guo et al. [2017], Müller et al. [2019], and adversely affects the generalization capability of the model. To mitigate this, training with probabilistic targets, also called soft targets, has shown to be beneficial Szegedy et al. [2016], Müller et al. [2019]. For instance, label smoothing injects noise into labels, leading to a higher entropy in the target conditional distribution Szegedy et al. [2016]. This was shown to be effective in a wide range of problems and, when combined with other soft target techniques Zhang et al. [2017], Han et al. [2022], leads to state-of-the-art performance Dosovitskiy et al. [2020], He et al. [2022].

Several avenues have been proposed to improve upon the log loss Gutmann and Hyvärinen [2010], Lin et al. [2017], including the InfoNCE objective Jozefowicz et al. [2016], Ma and Collins [2018]. InfoNCE has become the standard loss function for self-supervised contrastive learning due to its information theoretic properties, simplicity and excellent empirically-observed performance Chen et al. [2020], He et al. [2020], Tian et al. [2020], Oord et al. [2018]. It has shown to produce

Preprint. Under review.

versatile, high-quality representations that optimize a lower bound on their mutual information (MI) content Tian et al. [2020], Oord et al. [2018]. In natural language processing, InfoNCE was shown to be an efficient replacement for the log loss, with better generalization performance and on-par parameter estimation properties Jozefowicz et al. [2016], Ma and Collins [2018]. InfoNCE avoids the summation over all labels that is usually required to compute the normalization constant and instead relies on negative sampling. This can be advantageous when label spaces are large. Negative sampling has a strong connection to noise contrastive estimation as it ultimately estimates the true conditional $p(k|x)$ implicitly instead of using an explicit model Gutmann and Hyvärinen [2010], Ma and Collins [2018].

Despite its favourable theoretical properties as well as its empirical success in natural language processing and, more broadly, self-supervised learning, InfoNCE has not seen a wide adoption in supervised classification tasks involving other types of data such as images or graphs. One reason for this might be the difficulty of identifying scenarios in which InfoNCE would be a favourable loss function. Another reason might be that, due to its noise contrastive nature, combining it with powerful methods leveraging soft targets Szegedy et al. [2016], Zhang et al. [2017], Han et al. [2022], Yun et al. [2019] is not straightforward. This contrasts with the log loss, where soft target probabilities can be explicitly used to model the target conditional density. It is however conceivable that integrating such targets and their corresponding distributions might boost generalization performance.

In this work, we first show that InfoNCE is a relevant alternative to the log loss, particularly in the case of label uncertainty. We demonstrate through a toy example that InfoNCE indeed yields better parameter estimates in this scenario. This motivates us to suggest approaches that combine soft targets with noise contrastive estimation. Our main contribution is an InfoNCE generalization that directly integrates soft target probabilities into the loss by creating weighted linear combinations of label embeddings. Our loss function, which we dub soft target InfoNCE, can be derived using the continuous categorical distribution in combination with a Bayesian argument. We discuss the resulting attraction-repulsion dynamics as well as how this prior affects gradients. As second approach we suggest to sample targets from a distribution with higher entropy, such as the label smoothing distribution $p^\epsilon(y|x) = (1-\epsilon)p(y|x) + \epsilon\xi(y)$ in which $\epsilon \in [0, 1]$ determines the weighting between the underlying conditional and the noise distribution. Furthermore, we investigate how this affects the MI and entropy in the learned representation by deriving a variational lower bound.

Finally, we perform experiments on multiple classification benchmarks, including ImageNet, to compare and examine soft target InfoNCE's properties. Our results suggest that our loss is competitive with soft target cross-entropy and outperforms hard target NLL and InfoNCE. In an ablation study we give insights into the confidence calibration of the trained models and investigate how the number of negative samples/batch size as well as the amount of label smoothing noise affects model performance. Our results confirm the intuition that InfoNCE and soft target InfoNCE are important alternatives to the log loss and its soft variant. Our implementation is available at `https://github.com/uhlmanngroup/soft-target-InfoNCE.git`.

2 Background and related work

2.1 Log loss, soft targets and the continuous categorical distribution

In classification problems, the log loss, also known as negative log likelihood (NLL) or cross entropy loss, is an objective to evaluate the quality of probabilistic models Hastie et al. [2009]. It is the standard loss function used to train deep supervised classification models Krizhevsky et al. [2017], He et al. [2016], Dosovitskiy et al. [2020]. Intuitively, given a data sample, the log loss quantifies the discrepancy between the predicted class probability of the true label and the probability given through a target distribution. The most commonly used target is the degenerate conditional distribution $p(l|x) = \delta(l, k)$, induced by the one-hot encoding of labels into probabilities. The estimated probabilities q are usually computed as a softmax over class scores $q(k|x, \theta) := \exp(z \cdot w_k) / \sum_{l=1}^{K} \exp(z \cdot w_l)$, where k is the class label, z denotes the penultimate layer activations (also called embedding) of sample x, and w_k corresponds to the weights of the classifier layer responsible for computing the respective score. The log loss function can then be defined for sample x with label k as

$$l^{\mathrm{NLL}}(x, k) := -\log q(k|x, \theta). \tag{1}$$

A number of works have addressed different drawbacks of the log loss, including its robustness to noisy labels Zhang and Sabuncu [2018], Sukhbaatar et al. [2014] and poor margins Elsayed et al. [2018], Cao et al. [2019]. In addition to these shortcomings, neural networks trained with (1) tend to be poorly calibrated, which leads to overconfident predictions and to an increased misclassification rate Guo et al. [2017]. Besides scaling logits with a temperature parameter, label smoothing has shown to mitigate this problem Müller et al. [2019]. Label smoothing has been introduced as a model regularization technique in which the original "hard" one-hot encoding of a label is replaced by a "soft" target, computed as the weighted average of the original hard target and a noise distribution over labels, for instance the uniform distribution $\xi \equiv 1/K$ Szegedy et al. [2016]. Formally, the label smoothing distribution can be defined as

$$p^{\epsilon}(l|x) = (1-\epsilon)\delta(l,k) + \epsilon\xi(l), \tag{2}$$

where ξ denotes a noise distribution, $\epsilon \in [0,1]$, and where the delta distribution $\delta(l,k) = p(l|x)$ is the original target conditional distribution. During training, (1) is replaced by the soft target cross-entropy given by

$$l^{\mathrm{SoftTargetXent}}(x,k) = -\sum_{l=1}^{K} p^{\epsilon}(l|x)\log q(l|x,\theta). \tag{3}$$

By distributing small probability mass over labels that do not correspond to the true class, models are incentivised to reduce their prediction confidence, which has shown to improve performance Dosovitskiy et al. [2020]. Besides classification, soft targets are also used in knowledge distillation, in which a teacher network provides a student network its estimated probabilities as training targets Hinton et al. [2015].

The idea of distributing probability masses over two or more labels has also been used in combination with mixing input samples in a data augmentation technique called MixUp Zhang et al. [2017]. In MixUp data points, including their labels, are blend together into a new sample using a mixing coefficient, resulting in a new sample and a corresponding soft target. In this paper, we refer to targets that distribute probability mass among multiple classes as soft targets regardless of the underlying sample. This includes data augmentation techniques such as MixUp, as well as regularization methods like label smoothing.

Recently, Gordon-Rodriguez et al. [2020b] proposed a label smoothing loss based on the continuous categorical distribution Gordon-Rodriguez et al. [2020a] as an alternative to (3) that models non-categorical data more faithfully. The continuous categorical distribution generalizes its discrete analog and is defined on the closed simplex as

$$\alpha_1,\ldots,\alpha_K \sim \mathcal{CC}(\lambda) \iff p(\alpha_1,\ldots,\alpha_K;\lambda) \propto \prod_{i=1}^{K} \lambda_i^{\alpha_i}, \tag{4}$$

with $\sum_{i=1}^{K}\alpha_i = 1$.

2.2 InfoNCE for supervised classification

InfoNCE has been demonstrated to be a versatile loss function with applications across many machine learning tasks and application domains Tian et al. [2020], He et al. [2020], Poole et al. [2019], Damrich et al. [2022], Zimmermann et al. [2021]. In this work, we focus predominantly on its density estimation properties. In order to estimate densities, InfoNCE turns the problem into a supervised classification task in which the position of a single sample coming from the data distribution p must be identified in a tuple T of otherwise noise samples coming from a distribution η Jozefowicz et al. [2016]. The theoretical footing for conditional density estimation and classification has been introduced in the context of language modelling Ma and Collins [2018]. Concretely, Ma and Collins [2018] shows that InfoNCE's optimum θ^* are the parameters of the true conditional, *i.e.* $q(k|x,\theta^*) = p(k|x)$. In other words, supervised InfoNCE fits θ by minimizing the negative log likelihood of (x,k) being at a position S within T, given by $-\log P(S|T)$. Without loss of generality, for $S=0$, the InfoNCE loss can then be defined as

$$l^{\mathrm{InfoNCE}}(T) = -\log \frac{\exp(s(z,y_{0k_0})/\tau)}{\sum_{i=0}^{N}\exp(s(z,y_{ik_i})/\tau)}, \tag{5}$$

with $(x, y_{0k_0}) \sim p, y_{1k_1}, \ldots, y_{Nk_N} \sim \eta$, and where τ denotes the temperature and $s(z, y; \tau, \eta) = z^T y/\tau - \log \eta(y)$ is the scoring function. Further, y_{ik_i} is the learnable label embedding of class k_i, and i denotes the position of (x, k_i) within T. The label embeddings are a parametrized lookup table and are the analogs to the classifier scoring weights w_k of a neural network trained with the log loss.

Several variants of InfoNCE have been introduced, mostly in the context of self-supervised learning Chuang et al. [2020], Kalantidis et al. [2020], Chen et al. [2020], Li et al. [2023]. The supervised variant proposed in Khosla et al. [2020] is used for pre-training and leverages labels to create positive pairs. Further variants developed in Chuang et al. [2020], Li et al. [2023] correct for false negatives coming from the noise distribution by re-weighting the sum over negative samples, while Kalantidis et al. [2020] proposes to sample and create artificial hard negatives for training.

InfoNCE is also a biased estimator of MI Oord et al. [2018]. Maximizing the MI between learned representations that are based on related inputs (InfoMax principle Linsker [1988]) has yielded empirical success in representation learning Hjelm et al. [2018], Henaff [2020], Bachman et al. [2019], Chen et al. [2020], He et al. [2020], Veličković et al. [2018]. As MI is notoriously difficult to estimate, tractable variational lower bounds are usually optimized in practice Poole et al. [2019]. InfoNCE provides such a lower bound: more specifically, optimizing InfoNCE is equivalent to maximizing a lower bound of MI Oord et al. [2018], Tian et al. [2020].

3 Results

3.1 InfoNCE improves parameter estimation over negative log-likelihood in conditional models

Precise parameter estimation is essential for conditional models to obtain good generalization properties. Here we compare the estimation accuracy of InfoNCE against NLL in classification settings with different degrees of label uncertainty.

In its standard formulation, NLL makes the assumption that the target conditional distribution is degenerate, *i.e.* $p(l|x) = \delta(l, k)$ for a data sample x with class label k. However, in practice the true conditional distribution is often more complex, for instance due to uncertainty or ambiguity in the label assignment. As a consequence, an inherent amount of error is introduced into the estimated parameters. Several works have addressed this issue, for instance by introducing more entropy into the reference distribution Szegedy et al. [2016] or through alternative loss functions Gutmann and Hyvärinen [2010], including InfoNCE. InfoNCE does not require an explicit model as target distribution. Instead, parameters are estimated in an implicit manner by contrasting labels coming from the true distribution with labels coming from a noise distribution.

To study the benefits of this implicit approach, we consider cases in which the underlying conditional has different amounts of uncertainty (Figure Supp. 5). Specifically, we sample data points x from a mixture of Gaussians with K modes. Each mode $\theta_k \in \mathbb{R}^d$ corresponds to a class in the sense that labels for the data points x are sampled from the conditional model given by

$$p(k|x, \theta) = \frac{\exp(x^T \theta_k)}{\sum_{l=1}^{K} \exp(x^T \theta_l)}. \tag{6}$$

Concretely, we first sample $\mathcal{X} = \{x_1, \ldots, x_M\}$ from the mixture distribution and then subsample $\mathcal{X}$ into a dataset X of size $N > M$ with duplicates. Class labels are assigned by sampling from the conditional model $k \sim p(\cdot|x, \theta)$. We then estimate the parameters θ_k using NLL and InfoNCE. We increase the uncertainty by increasing the degree of alignment between modes θ_k, as detailed in Supplementary Section A.1.

The results, visualized in Figure 1, indicate that increased levels of label uncertainty lead to worse parameter estimates using NLL when compared to InfoNCE. As anticipated, parameter estimation accuracy deteriorates as the mode alignment degree increases for both loss functions. However, the error of InfoNCE's estimates relative to those of NLL increases considerably more slowly. Our experiment demonstrates the advantage of implicit sampling from a dataset generated by the true data distribution as opposed to using an explicit model distribution as target.

In order to connect this result with deep neural networks, one can interpret the data X as embeddings (*i.e.* penultimate layer activations) and the θ_k as the classification weights producing the logits of

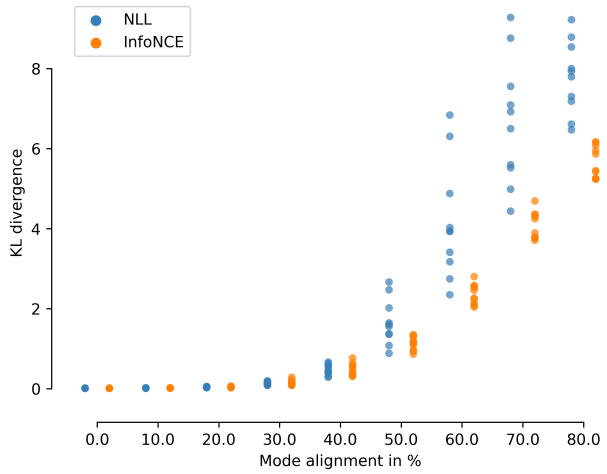


Figure 1: Parameter estimation quality of InfoNCE vs. negative log-likelihood in conditional density models with different degrees of mode alignment. The estimation error is reported as the KL divergence. The alignment degree corresponds to the angle between modes. 0% alignment corresponds to orthogonal modes and 80% corresponds to an angle of $18°$.

the last layer. For orthogonal modes (0% alignment), X is linearly separable and both objectives give comparable estimates. However, for non-orthogonal modes (alignment $> 0\%$), InfoNCE might lead to better generalization. This makes our observations particularly relevant in fine-tuning settings where only the last layer is re-trained.

3.2 InfoNCE with soft targets

Motivated by InfoNCE's favourable parameter estimation properties and the empirical success of training with soft targets, we sought to combine both. In this section, we propose a principled, conceptually simple loss function that integrates soft targets directly into InfoNCE.

Derivation. Consider the continuous categorical distributions $\mathcal{CC}(p_{Y|X}(\cdot|x)), \mathcal{CC}(\eta)$ induced by the true conditional and noise distribution, respectively.[1] Let $T = (\alpha_1, \dots, \alpha_{N+1})$ be a tuple of length $N+1$, with $\alpha_i \geq 0$, $\sum_{l=1}^{K} \alpha_{il} = 1$, containing a single sample $\alpha \sim \mathcal{CC}(p_{Y|X}(\cdot|x))$ and noise samples $\tilde{\alpha} \sim \mathcal{CC}(\eta)$ otherwise. We are interested in identifying the position S of α given the tuple T, *i.e.* $P(S|T)$. A priori, we have $P(S = i) = 1/(N+1)$ for all allowed positions $i = 1, \dots, N+1$. Further, we know that $P(\alpha_j|S = k) = p_{\mathcal{CC}}(\alpha_j; \eta)$ for all $j \neq k$, and $P(\alpha_k|S = k) = p_{\mathcal{CC}}(\alpha_k; p_{Y|X}(\cdot|x))$. Thus,

$$P(T|S = k) = p_{\mathcal{CC}}(\alpha_k; p_{Y|X}(\cdot|x)) \prod_{\substack{i=1 \\ i \neq k}}^{N+1} p_{\mathcal{CC}}(\alpha_i; \eta) = \frac{p_{\mathcal{CC}}(\alpha_k; p_{Y|X}(\cdot|x))}{p_{\mathcal{CC}}(\alpha_k; \eta)} \prod_{i=1}^{N+1} p_{\mathcal{CC}}(\alpha_i; \eta). \quad (7)$$

From this, we can infer the marginal

$$P(T) = \frac{1}{N+1} \prod_{i=1}^{N+1} p_{\mathcal{CC}}(\alpha_i; \eta) \sum_{k=1}^{N+1} \frac{p_{\mathcal{CC}}(\alpha_k; p_{Y|X}(\cdot|x))}{p_{\mathcal{CC}}(\alpha_k; \eta)}. \quad (8)$$

By applying Bayes' theorem in combination with (7) and (8), we can compute the posterior

$$P(S = k|T) = \frac{\frac{p_{\mathcal{CC}}(\alpha_k; p_{Y|X}(\cdot|x))}{p_{\mathcal{CC}}(\alpha_k; \eta)}}{\sum_{l=1}^{N+1} \frac{p_{\mathcal{CC}}(\alpha_l; p_{Y|X}(\cdot|x))}{p_{\mathcal{CC}}(\alpha_l; \eta)}} = \frac{\prod_{i=1}^{K} \left(\frac{p_{Y|X}(i|x)}{\eta(i)} \right)^{\alpha_{ki}}}{\sum_{l=1}^{N+1} \prod_{j=1}^{K} \left(\frac{p(_{Y|X}(j|x)}{\eta(j)} \right)^{\alpha_{lj}}}. \quad (9)$$

[1] To simplify notation we denote $\mathcal{CC}([p_{Y|X}(1|x), \dots, p_{Y|X}(K|x)])$ as $\mathcal{CC}(p_{Y|X}(\cdot|x))$ and $\mathcal{CC}([\eta(1), \dots, \eta(K)])$ as $\mathcal{CC}(\eta)$

We estimate the density ratio $p_{Y|X}(i|x)/\eta(i)$ using the parametric model $\exp(s(z, y_k; \tau, \eta))$ where $s(z, y_k; \tau, \eta) = z^T y_k/\tau - \log \eta(y)$ and $z = f(x; \theta)$, which can for instance be a neural network. By inserting this back into (9), we can state the maximum likelihood estimation problem

$$\max_{\theta} \frac{\exp\left(\sum_{i=1}^{K} \alpha_{ki} s(z, y_i; \tau, \eta)\right)}{\sum_{l=1}^{N+1} \exp\left(\sum_{j=1}^{K} \alpha_{lj} s(z, y_j; \tau, \eta)\right)}. \tag{10}$$

Taking the negative logarithm, we can rewrite (10) as our soft target InfoNCE loss function

$$\mathop{\mathbb{E}}_{\substack{\alpha_k \sim \mathcal{CC}(p_{Y|X}) \\ \alpha_1, \ldots, \alpha_{k-1}, \alpha_{k+1}, \ldots, \alpha_{N+1} \sim \mathcal{CC}(\eta)}} \left[-\log \frac{\exp\left(\sum_{i=1}^{K} \alpha_{ki} s(z, y_i; \tau, \eta)\right)}{\sum_{l=1}^{N+1} \exp\left(\sum_{j=1}^{K} \alpha_{lj} s(z, y_j; \tau, \eta)\right)} \right]. \tag{11}$$

Since $\exp(\sum_{i=1}^{K} \alpha_{ki} s(z, y_i; \tau, \eta))$ converges to the density ratio $p(\alpha_l; p_{Y|X})/p(\alpha_l; \eta)$, we can simply use $\exp(z^T y_i/\tau)$ during inference to estimate a proportional score of the probability $p_{Y|X}(i|x)$.

Discussion. To provide an intuition for (11), we restate it as a negative temperature scaled energy cross-entropy

$$-\sum_{i=1}^{K} \alpha_{ki} \log \frac{\exp\left(s(z, y_i; \tau, \eta)\right)}{\sum_{l=1}^{N+1} \exp\left(\sum_{j=1}^{K} \alpha_{lj} s(z, y_j; \tau, \eta)\right)}, \tag{12}$$

where the fraction inside of the logarithm can be interpreted as an energy function[2]. Notably, (12) is structurally similar to the soft target cross-entropy (3). For each class, we now have an attractive pull acting on its embedding y_i and z weighted by α_{ki}. For instance, in case of label smoothing with a uniform noise distribution $\xi \equiv 1/K$, we have $\alpha_{ki} = 1 - (K - 1/K)\epsilon$ for the true class i and $\alpha_{kj} = \epsilon/K$ otherwise, which results in a dampened attraction between the data embedding and the original hard target.

While the inference model $\exp(z^T y_i/\tau)$ from the original InfoNCE does not change, the gradient dynamics induced by the respective energy landscapes are different. A more detailed discussion of the change in attraction-repulsion dynamics as well as a derivation for the gradient is provided in the Supplementary Section A.2.

3.3 InfoNCE with soft distributions

An alternative to integrating soft weights directly into the loss function is to sample from the corresponding distribution instead. In this section, we examine an InfoNCE loss in which labels are drawn from the soft distribution

$$p^{\epsilon}(k|x) = (1 - \epsilon)p(k|x) + \epsilon\xi(k), \tag{13}$$

instead of the data generated by the true conditional. A derivation of this loss is provided in the Supplementary Section A.3 and follows directly from the standard derivation of InfoNCE (e.g. see Damrich et al. [2022]).

Effect of soft distributions on the MI lower bound. Here, we sketch the derivation and discuss the implications of sampling from the soft distribution (13) on InfoNCE's MI lower bound with respect to the underlying data distribution p. We provide a complete proof in Supplementary A.3.

As we saw in the previous section, the optimal critic is proportional to the density ratio between the soft data and the noise distribution

$$\exp(s(z, y)/\tau) \propto \frac{p^{\epsilon}_{Y|Z}(y|z)}{p_Y(y)}, \tag{14}$$

[2]Note that $q(k|z) = \exp(s(z, y_i))/\sum_{l=1}^{N+1} \exp(\sum_{j=1}^{K} \alpha_{lj} s(z, y_j))$ is not necessarily a probability since $\sum_{k=1}^{K} q(k|x)$ does not sum to 1 in general.

with $\eta \equiv \xi \equiv p_Y$. Using the argument provided in the proof of the MI lower bound Tian et al. [2020], Oord et al. [2018], we directly obtain

$$l^{\text{SDInfoNCE}}(T) \geq \log N - \mathop{\mathbb{E}}_{(y,z)\sim p^\epsilon} \left[\log \frac{p^\epsilon_{Y|Z}(y|z)}{p_Y(y)} \right]. \tag{15}$$

The expected value can be written as

$$\mathop{\mathbb{E}}_{(y,z)\sim p^\epsilon} \left[\log \frac{p^\epsilon_{Y|Z}(y|z)}{p_Y(y)} \right] = I(Z;Y) + H(Y|Z) - H_\epsilon(Y|Z), \tag{16}$$

where $H(Y|Z)$ denotes the conditional entropy with respect to $p_{Y,Z}$ and

$$H_\epsilon(Y|Z) = \mathop{\mathbb{E}}_{(y,z)\sim p^\epsilon_{Y,Z}} \left[-\log p^\epsilon_{Y|Z}(y|z) \right]. \tag{17}$$

Inserting this back into (15) yields following inequality

$$I(Z;Y) + H(Y|Z) - H_\epsilon(Y|Z) \geq \log N - l^{\text{SDInfoNCE}}(T). \tag{18}$$

For $\epsilon = 0$, we have $H_0(Y|Z) = H(Y|Z)$ and the above bound returns to its original form. In contrast, for $\epsilon = 1$, $H_1(Y|Z) = H(Y) = I(Y;Z) + H(Y|Z)$, (18) becomes trivial due to the high entropy in $p^{\epsilon=1}$.

The left hand side of (18) is larger or equal to zero (see Supp. A.3). Thus, by minimizing InfoNCE when sampling from a soft distribution, the baseline MI as well as the entropy increase. Intuitively, ϵ regulates the amount of entropy and MI distilled into the learned representations. High values can increase prediction uncertainty, which may help with ambiguous instances and model calibration, while low values boost model confidence, and increase MI between input and class embeddings.

4 Experiments

We show that soft target InfoNCE achieves competitive results compared to soft target cross-entropy and surpasses NLL, InfoNCE and soft distribution InfoNCE in quantitative evaluation on classification benchmarks, including ImageNet. We provide an ablation study on the sensitivity to varying numbers of noise samples, varying amounts of label smoothing noise and examine calibration properties, in order to obtain further insights into the models trained with our loss.

Loss implementation. We provide a simple implementation of (soft target) InfoNCE that directly works on class scores (logits) and, thus, is compatible with standard classification models (Figure 2). InfoNCE loss implementations usually take as inputs embedding vectors He et al. [2020], Chen et al. [2020], Khosla et al. [2020], requiring additional code adaptations of the neural network in order to work in the supervised classification case. More specfically, in order to compute class scores (and soft target embeddings), classification head weights need to be accessed. While this is inconvenient, it also creates discrepancies between training and inference code. Instead, we leverage soft targets or one-hot encodings, in case of hard targets, to directly work on the logits.

4.1 Classification accuracy

We perform image classification experiments on ImageNet Krizhevsky et al. [2012], Tiny ImageNet Krizhevsky et al. [2009] and CIFAR-100 Krizhevsky et al. [2009], and node classification experiments on the CellTypeGraph benchmark Cerrone et al. [2022]. For image classification we use a vision transformer base (ViT-B/16) Dosovitskiy et al. [2020] with a 3-layer MLP classification head, pre-trained as a masked autoencoder He et al. [2022] with image patches of size 16×16. For node classification, we use a DeeperGCN Li et al. [2019, 2020] and report performance on this task as the average over a 5-fold cross- validation experiment Cerrone et al. [2022].

For the experiments to be comparable, we use the same training recipe and only adapt the (base) learning rate for optimal performance of the respective loss function. Specifically, we use the same neural network architecture, fix the temperature to $\tau = 1.0$, and use the dot-product as scoring function $s(z, y) = z^T y$. For image classification, we follow common practice for ViT end-to-end

```
class SoftTargetInfoNCE(nn.Module):
def __init__(self, noise_probs, T=1.0):
    # noise_probs: (K,) noise probabilities over classes
    # T: temperature
    super().__init__()
    self.K = noise_probs.shape[0]
    self.log_noise = torch.log(noise_probs).unsqueeze(0)
    self.T = T

def forward(self, logits, targets):
    # logits: (N, K) class scores (not true logits!)
    # targets: (N, K) (soft) target weights for each class
    N = targets.shape[0]
    targets = targets.repeat(N,1).unsqueeze(1).reshape(N, N, self.K)
    logits = (logits / self.T) - self.log_noise
    logits = (logits.unsqueeze(1) * targets).sum(-1)
    logits -= (torch.max(logits, dim=1, keepdim=True)[0]).detach()
    labels = torch.arange(N, dtype=torch.long).cuda()
    return nn.CrossEntropyLoss()(logits, labels)
```

Figure 2: Soft target InfoNCE PyTorch code. For clarity, this assumes training on one GPU. In the distributed case, this has to be slightly adapted in order to gather (soft) targets from other GPUs such that they can be used as additional negatives. For more details see code repository.

fine-tuning, using the recipe provided in He et al. [2022]. We train ViTs for 100 epochs with a batch size of 1024 using label smoothing $\epsilon = 0.1$, MixUp $\alpha_M = 0.8$ and CutMix of $\alpha_C = 1.0$. For node classification, we train a 32-layer DeeperGCN with a batch size of 8 and a learning rate of 0.001. The hard target cross-entropy and InfoNCE baselines are trained for 500 epochs due to overfitting, while the soft target baselines are trained for 1000 epochs with label smoothing of $\epsilon = 0.1$. Further technical details are provided in the Supp. A.4.

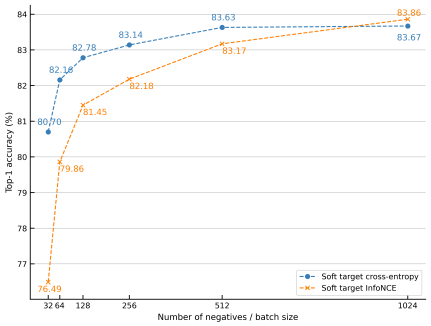
Comparison to soft target cross-entropy. Table 1 shows a comparison of soft target InfoNCE to its cross-entropy loss analog across the four benchmark datasets. While soft target cross-entropy gives slightly better results on ImageNet ($\Delta 0.29\%$) and CellTypeGraph ($\Delta 0.55\%$), we obtain on-par performance on Tiny ImageNet ($\Delta 0.19\%$) and CIFAR-100 ($\Delta 0.06\%$). More broadly, we observe similar convergence behaviour and performance for both loss functions as a response to learning rates within the same range. For instance, for image classification, we found that base learning rates between 0.0002 and 0.0006 lead to optimal results in both cases. Notably, for image classification, we chose a 3 layer MLP as classification head over a linear layer since we observed gains in top-1 accuracy for both loss functions. However, gains for soft target InfoNCE were consistently larger ($\Delta 1.72\%$ Tiny ImageNet, $\Delta 0.7\%$ CIFAR-100) compared to soft target cross-entropy (Tiny ImageNet $\Delta 0.6\%$, CIFAR-100 $\Delta 0.36\%$) not only in top-1 performance but also across runs. Since performance seems to saturate around the same accuracy levels, we conclude that soft target InfoNCE is competitive with soft target cross-entropy.

Comparison to InfoNCE with soft distribution sampling. Table 1 compares soft target InfoNCE against the more simplistic approach of sampling a hard target from the distribution provided by the soft targets. Integrating soft targets directly into InfoNCE consistently results in better top-1 accuracy across all benchmarks. However, sampling from a soft distribution results in the worst performing models on CIFAR-100 and CellTypeGraph and is only on-par with the hard target baselines.

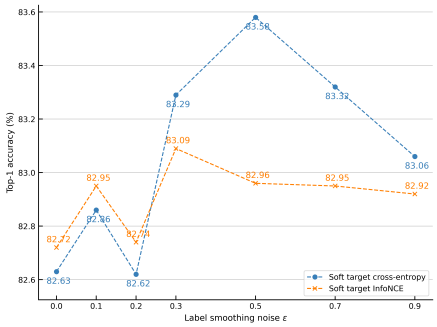
Comparison to hard target baselines. We finally compare soft target InfoNCE to the standard cross-entropy loss (NLL) and InfoNCE, which both leverage hard targets. Regularization techniques such as label smoothing and MixUp, thus, cannot be used in combination with these baselines. While NLL is on-par with soft target InfoNCE on CIFAR-100 ($\Delta 0.04\%$) and CellTypeGraph ($\Delta 0.2\%$), it is outperformed on ImageNet ($\Delta 1.19\%$) and TinyImageNet ($\Delta 1.23\%$). This also holds for the InfoNCE baseline, which has the same performance pattern as the NLL baseline.

Table 1: Top-1 classification accuracy for various datasets. We compare soft target InfoNCE to different baseline loss functions.

	ImageNet	Tiny ImageNet	CIFAR-100	CellTypeGraph
NLL	82.35	82.63	**90.84**	86.92
Soft target cross-entropy	**83.85**	83.67	90.74	**87.67**
InfoNCE	82.52	82.72	90.75	86.80
Soft distribution InfoNCE	82.96	82.77	89.82	86.62
Soft target InfoNCE	83.54	**83.86**	90.80	87.12



(a) Top-1 accuracy as a function of noise samples/batch size for ViT-B/16 models trained with soft target losses in combination with label smoothing, MixUp and CutMix.



(b) Top-1 accuracy as a function of label smoothing noise for ViT-B/16 models trained with soft target losses.

Figure 3: Noise sample (left) and label smoothing (right) ablation experiments on Tiny ImageNet.

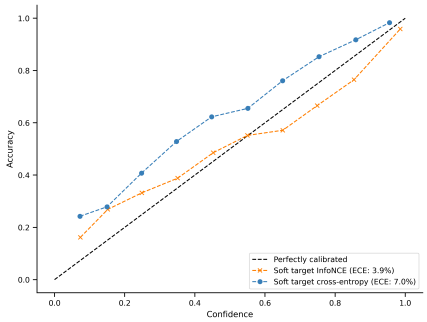
4.2 Ablation experiments

In our ablation experiments, we explore different aspects of the soft target InfoNCE loss using the ViT-B/16 model described in the previous section.

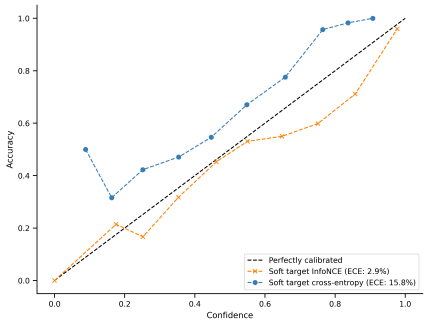
Number of noise samples. Soft target InfoNCE relies on noise samples to contrast the true soft target with. While this can be separated from the batch size, it is more efficient to couple both and leverage other soft targets as noise samples. We give a soft target cross-entropy baseline for comparison. Figure 3 visualizes the results of this experiment in terms their top-1 performance on Tiny ImageNet. As anticipated, smaller batch sizes have a stronger negative effect on performance for models trained with the noise contrastive loss compared to the baseline, due to fewer available negative samples, resulting in a performance difference of $\Delta 4.21\%$ for the smallest batch size. However, as the batch size increases ViTs trained with soft target InfoNCE are on-par with their cross-entropy analog, and slightly outperform them by $\Delta 0.19\%$.

Sensitivity to label smoothing. We investigate how different amounts of label smoothing noise ϵ affect models trained with soft target InfoNCE. Figure 3 visualizes the results in terms of their top-1 accuracy on Tiny ImageNet. Notably, accuracy of soft target InfoNCE models varies less (max. deviation of $\Delta 0.37\%$ and standard deviation of 0.12) compared to the soft target cross-entropy baseline (max deviation of $\Delta 0.96\%$ and standard deviation of 0.34).

Confidence calibration. Label smoothing in combination with cross-entropy has shown to improve model calibration Müller et al. [2019]. Here, we investigate confidence calibration properties of our vision transformer models trained with soft target InfoNCE and compare to its cross-entropy analog. Since InfoNCE models converge to unnormalized density functions for which standard calibration metrics can not be computed, we treat the model outputs as logits and normalize them using a softmax function. Figure 4 shows reliability diagrams of ViTs trained on Tiny ImageNet with label smoothing $\epsilon = 0.1$ and of the best performing ViTs on CIFAR-100 as described in the previous section. Notably, models trained with soft target InfoNCE were better calibrated in these two cases. Specifically, we



(a) Soft target losses trained with label smoothing on Tiny ImageNet.



(b) Best performing models on CIFAR-100 trained with soft target losses in combination with label smoothing, MixUP and CutMix.

Figure 4: Reliability diagrams of ViT-B/16 trained on Tiny ImageNet (left) and CIFAR-100 (right).

obtained an expected calibration error (ECE) of 3.9% on Tiny ImageNet and 2.9% on CIFAR-100 which was lower than the respective ECE of the soft target cross-entropy models (7% Tiny ImageNet, 15.8% CIFAR-100). We additionally compared the best performing Tiny ImageNet models of the previous section for which we obtained a slightly worse ECE of 7.7% for the soft target InfoNCE ViT compared to an ECE of 6.3% for the soft target cross-entropy ViT (Figure A.4).

5 Conclusions and limitations

We presented soft target InfoNCE, a principled generalization of InfoNCE for conditional density estimation in supervised classification problems. We derived our loss from first principles using a continuous categorical distribution in combination with a Bayesian argument. The ability to leverage soft targets coming from sophisticated augmentation and regularization techniques allows our loss function be on-par with soft target cross-entropy and to outperform hard target NLL and InfoNCE losses.

As limitations we identified the drop in performance when a small batch size is used in combination with mini-batch and noise sample coupling. However, we leave it up to future work to investigate if this can be mitigated with separately sampled soft noise targets. Further, we observed a slight performance drop with linear classification heads when compared to soft target cross-entropy. Nevertheless, deep classification models are typically optimized to work well with cross-entropy, and we observed competitive performance when employing a 3-layer MLP classification head. This suggests that by exploring different neural architectures one might find more fitting alternatives.

In additional future work we want to explore different noise distributions for sampling soft weights as well as test our loss in other problem settings such as knowledge distillation.

6 Acknowledgements

The authors thank Anna Kreshuk (EMBL) for many valuable exchanges and insightful discussions.

References

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Lorenzo Cerrone, Athul Vijayan, Tejasvinee Mody, Kay Schneitz, and Fred A Hamprecht. Celltype-graph: A new geometric computer vision benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20897–20907, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Sebastian Damrich, Jan Niklas Böhm, Fred A Hamprecht, and Dmitry Kobak. Contrastive learning unifies t-sne and umap. *arXiv preprint arXiv:2206.01816*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gamaleldin Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. *Advances in neural information processing systems*, 31, 2018.

Elliott Gordon-Rodriguez, Gabriel Loaiza-Ganem, and John Cunningham. The continuous categorical: a novel simplex-valued exponential family. In *International Conference on Machine Learning*, pages 3637–3647. PMLR, 2020a.

Elliott Gordon-Rodriguez, Gabriel Loaiza-Ganem, Geoff Pleiss, and John Patrick Cunningham. Uses and abuses of the cross-entropy loss: Case studies in modern deep learning. *arXiv preprint arXiv:2011.05231*, 2020b.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

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. JMLR Workshop and Conference Proceedings, 2010.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, pages 8230–8248. PMLR, 2022.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. *Advances in Neural Information Processing Systems*, 33: 21798–21809, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *University of Toronto*, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Guohao Li, Matthias Müller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*, 2020.

Haochen Li, Xin Zhou, Luu Anh Tuan, and Chunyan Miao. Rethinking negative pairs in code search. *arXiv preprint arXiv:2310.08069*, 2023.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

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

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812*, 2018.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

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

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020.

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *International Conference on Machine Learning*, pages 12979–12990. PMLR, 2021.

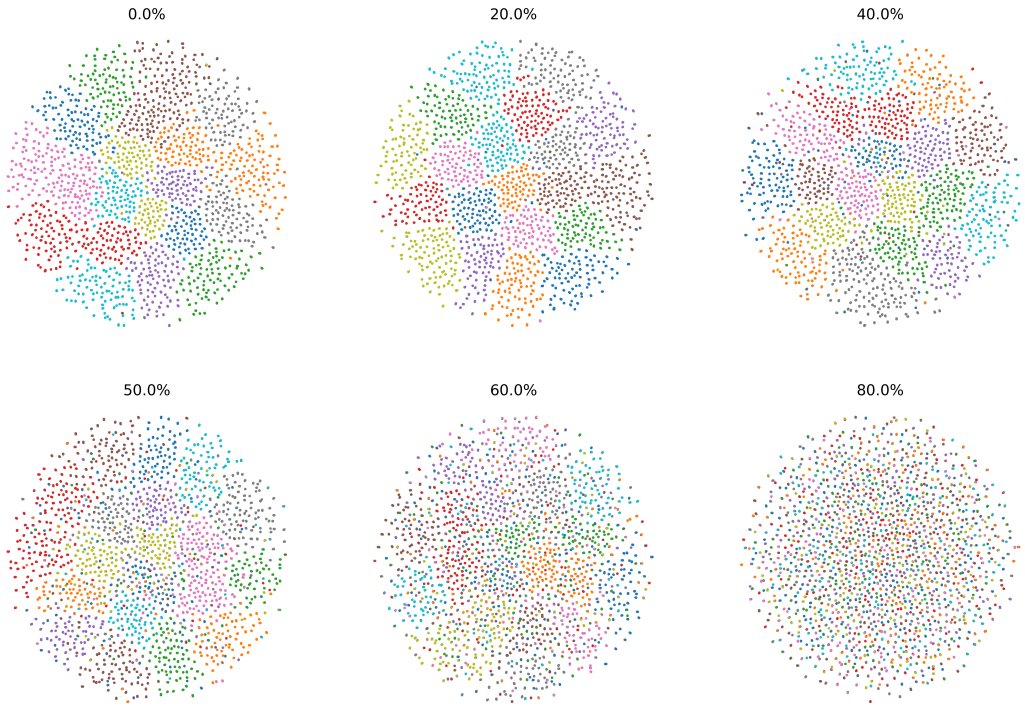


Figure 5: Different degrees of mode alignments of datasets X visualized using TSNE. The alignment degree corresponds to the angle between modes. 0% alignment corresponds to orthogonal modes and 80% corresponds to an angle of $18°$.

A Supplementary material: Noise contrastive estimation with soft targets for conditional models

A.1 InfoNCE improves parameter estimation over negative log-likelihood in conditional models

In order to compare parameter estimation quality of InfoNCE and NLL, we create nine datasets sampled from different Gaussian mixture models with different degrees of mode alignment. Specifically, we sample each dataset $\mathcal{X}$ from a GMM $p(x) = \sum_{i=1}^{20} \pi_i \mathcal{N}(x|10 \cdot \theta_i, I)$, with $\theta_i \in \mathbb{R}^{20}, \|\theta_i\| = 1$ and $\pi_i \equiv 1/20$. The amount of mode alignment in $\%$ in a GMM corresponds to the scaling factor applied to $\pi/2$ determining the angles $\phi_1, \ldots, \phi_{19}$ of the 20-spherical coordinate system. Concretely, let s be a scaling factor determined by the percentage according to $s = 100/(100 - \text{perc.})$ then ϕ is computed as $\phi = \pi/(2s)$. Intuitively, we can interpret this as fixing one mode in an orthogonal system and contracting/aligning the remaining modes to the fixed one.

For each experiment we create a dataset $\mathcal{X}$ composed of 1600 data points sampled from a GMM. $\mathcal{X}$ is then uniformly subsampled into X with $|X| = 32000$, and we assign labels to each $x \in X$ based on the conditional model $p(k|x, \theta) = \exp(x^T \theta_k) / \sum_{l=1}^{K} \exp(x^T \theta_l)$ described in the main text.

For fitting the parameters, we train for 500 epochs with early stopping, a batch size of 1024 and a learning rate of 0.001. We evaluate parameter estimation quality for each GMM over 10 random seeds.

Figure 5 visualizes datasets X with different degrees of mode alignment.

A.2 InfoNCE with soft targets: discussion

Attraction-repulsion in

$$l = -\sum_{i=1}^{K} \alpha_{ki} \log \frac{\exp\left(s(z, y_i; \tau, \eta)\right)}{\sum_{l=1}^{N+1} \exp\left(\sum_{j=1}^{K} \alpha_{lj} s(z, y_j; \tau, \eta)\right)}, \tag{19}$$

can be further separated by restating the loss function as the two summands

$$-\sum_{i=1}^{K} \alpha_{ki} \log \frac{\exp\left(s(z, y_i)\right)}{\exp(\alpha_{ki} s(z, y_k)) + \sum_{\substack{l=1 \\ l \neq k}}^{N+1} \exp\left(\alpha_{li} s(z, y_i) + \sum_{\substack{j=1 \\ j \neq i}}^{K} \tilde{\alpha}_{lj} s(z, y_j)\right)}, \tag{20}$$

with $\tilde{\alpha}_{lj} = \alpha_{lj} - \alpha_{ki}$, and

$$\sum_{i=1}^{K} \alpha_{ki} \sum_{\substack{l=1 \\ l \neq k}}^{K} \alpha_{li} s(z, y_l), \tag{21}$$

where we have omitted the dependence of the scoring function s on τ and η for readability. Similar to (19), (20) predominantly contributes with an attractive pull between the label embedding y_i and the data embedding z that is regulated by α_{ki}, due to the numerator in the logarithm. We discuss the denominator as well as the second term (21) in the context of label smoothing with a uniform noise distribution $\xi \equiv 1/K$, i.e. $\alpha_{ki_0} = 1 - (K-1)\epsilon/K$ for the true class i_0 and $\alpha_{kj} = \epsilon/K$ for $j \neq i_0$ otherwise. For the composition of noise labels, we differentiate between the following cases:

- In case of $i = i_0$ and the l-th sample has label $i_l \neq i_0$, we have $\alpha_{li} = \epsilon/K, \tilde{\alpha}_{li_l} = 1 - \epsilon$, and $\tilde{\alpha}_{lj} = 0$ otherwise. Effectively this acts as repulsion between z and label i_0 of strength ϵ/K as well as z and label i_l of strength $1 - \epsilon$.
- In case of $i = i_0$ and the l-th sample has label $i_l = i_0$, we have $\alpha_{li_0} = 1 - (K-1)\epsilon/K$, and $\tilde{\alpha}_{lj} = 0$, resulting in a repulsion between z and i_0.
- In case of $i \neq i_0$ and the l-th sample has label $i_l \neq i_0$, we have $\alpha_{li_0} = \epsilon - 1$ which results in an attractive force between z and label i_0. Further, in case of $i_l = i$ then $\alpha_{li} = 1 - (K-1)\epsilon/K$, and $\tilde{\alpha}_{lj} = 0$ otherwise. In the event of $i_l \neq i$, we have $\alpha_{li} = \epsilon/K, \tilde{\alpha}_{li_l} = 1 - \epsilon$, and $\tilde{\alpha}_{lj} = 0$ otherwise. Ultimately, this acts as a repulsion between z and the labels i, i_l with the respective strength.
- In case of $i \neq i_0$ and the l-th sample has label $i_l = i_0$, we have $\alpha_{li} = \epsilon/K$ and $\tilde{\alpha}_{lj} = 0$ otherwise, which results in a repulsive force between z and label i.

The second term (21) is a penalty term and acts as a repulsion between z and all classes while penalzing the true class i_0 less than others in case of $\epsilon < 1/2$. In summary, soft targets change the attraction repulsion dynamics between an embedded data point and all label embeddings by adjusting the amount of pull-push applied.

Gradient analysis. Here, we derive and examine the gradient responses of soft target InfoNCE with respect to data embeddings z. We consider as scoring function the cosine similarity (with temperature $\tau = 1$) $s(z, y) = z^T y / \|z\| \|y\|$ and first compute the gradient with respect to $\hat{z} = z/\|z\|$.

$$\partial_{\hat{z}} l = -\left(\sum_{i=1}^{K} \alpha_{ki} y_i - \sum_{l=1}^{N+1} C_l(z, y) \sum_{j=1}^{K} \alpha_{lj} y_j\right) \tag{22}$$

$$= \sum_{i=1}^{K} \left(-\alpha_{ki} y_i + \sum_{l=1}^{N+1} C_l(z, y) \alpha_{lj}\right) y_j, \tag{23}$$

with $C_l(z, y) = \exp\left(\sum_{j=1}^{K} \alpha_{lj} s(z, y_j)\right) / \sum_{l=1}^{N+1} \exp\left(\sum_{j=1}^{K} \alpha_{lj} s(z, y_j)\right)$. Thus, for the unnormalized embeddings z, we have

$$\partial_z l = \frac{I - \hat{z}\hat{z}^T}{\|z\|} \partial_{\hat{z}} l \tag{24}$$

For soft target embeddings, our loss function inherits gradient response properties directly from the original InfoNCE. Thus, for weak targets[3] gradient magnitudes approach zero, while for strong targets[4] gradient magnitudes are larger than zero.

For the original hard targets, we obtain gradient magnitudes larger than zero when the target is strong, *i.e.* when $z^T \cdot y_j \approx 0$. More specifically, we would obtain the following gradient response with respect to y_j:

$$\alpha_{kj} \left\| -1 + \frac{\sum_{l=1}^{N+1} \alpha_{lj} \exp\left(\sum_{i=1}^{K} \alpha_{li} s(z, y_i)\right)}{\sum_{i=1}^{K} \alpha_{kj} \exp\left(\sum_{i=1}^{K} \alpha_{li} s(z, y_i)\right)} \right\|. \tag{25}$$

As in the previous discussion, we consider the case of label smoothing with a uniform noise distribution. In this case, for $\epsilon \in (0, 1/2)$ the second sum in the denominator is smaller than 1 since $\alpha_{lj}/\alpha_{kj} \leq 1$, and larger otherwise. Thus, in both cases the term in (25) is larger than zero.

The analytic expression of soft target InfoNCE gradients is similar to those of soft target cross-entropy, although with distinct characteristics: soft target InfoNCE involves a summation over batch sample embeddings instead of classes, and additionally, the soft labels directly impact the logits. The analytical expressions we derived for the gradient magnitudes for scenarios involving hard positive and hard negative pairs reveal that in the presence of hard positives, the gradient magnitude increases with the number and strength of the negative pairs. Similarly, in instances in which hard negatives are given, the gradient magnitude increases proportionally with the strength of the positive as well as the number and strength of the negative pairs.

A.3 InfoNCE with soft distributions

Derivation. Let S be the random variable denoting the position of y_0 within T. We are interested in recovering the position S given the knowledge of T, *i.e.* $P(S|T)$. A priori, we have $P(S = i) = 1/(N + 1)$ for all allowed positions $i = 0, \ldots, N$. Further, we know that $P(y_j|S = 0) = \eta(y_j)$ for all $j \neq 0$, and $P(y_0|S = 0) = p^\epsilon(y_0)$. Thus,

$$P(T|S = 0) = \frac{p^\epsilon(y_0|x)}{\eta(y_0)} \prod_{i=0}^{N} \eta(y_i). \tag{26}$$

From this, we can infer

$$P(T) = \frac{1}{N+1} \prod_{i=0}^{N} \eta(y_i) \sum_{k=0}^{N} \frac{p^\epsilon(y_k|x)}{\eta(y_k)}. \tag{27}$$

By applying Bayes' theorem in combination with (26) and (27), we can compute the posterior

$$P(S = 0|T) = \frac{\frac{p^\epsilon(y_0|x)}{\eta(y_0)}}{\sum_{k=0}^{N} \frac{p^\epsilon(y_k|x)}{\eta(y_k)}}. \tag{28}$$

We estimate the ratio $p^\epsilon(y|x)/\eta(y)$ using the parameterized model

$$\exp(s(z, y)/\tau), \tag{29}$$

where $z = f(x; \theta)$ is for instance a neural network. Inserting (29) into (28) and taking the negative logarithm results in the soft distribution InfoNCE loss function

$$l^{\text{SDInfoNCE}}(T) = -\log \frac{\exp(s(z, y_0)/\tau)}{\sum_{k=0}^{N} \exp(s(z, y_k)/\tau)}. \tag{30}$$

Notably, for $\eta \equiv \xi$, the parameterized model (29) converges to a function proportional to the density ratio

$$(1 - \epsilon)\frac{p(y|x)}{\xi(y)} + \epsilon. \tag{31}$$

Multiplying by $\xi(y)$ allows to recover the density $(1 - \epsilon)p(y|x) + \epsilon\xi(y)$.

[3]These are targets with $z^T \cdot y \approx 1$ in the case of true targets, and $z^T \cdot y \approx -1$ in the case of noise or negative targets.

[4]These are targets with $z^T \cdot y \approx 0$.

Effect of soft distributions on the MI lower bound: derivation. Here we derive a variational lower bound for soft distribution InfoNCE connecting MI and entropy properties of the learned representations. To recapitulate, let $p_{Y,Z}$ be the "data" distribution, p_Y the negative sampling distribution as well as the noise distribution for label smoothing, *i.e.* $p_Y \equiv \eta \equiv \xi$. Further, we define the conditional and joint soft data distributions as

$$p^{\epsilon}_{Y|Z}(y|z) = (1-\epsilon)p_{Y|Z}(y|z) + \epsilon p_Y(y), \tag{32}$$

$$p^{\epsilon}_{Z,Y}(z,y) = (1-\epsilon)p_{Z,Y}(z,y) + \epsilon p_Y(y)p_Z(z). \tag{33}$$

Next, we derive the lower bound

$$l^{\text{SDInfoNCE}}(T) \geq \mathop{\mathbb{E}}_{T}\left[-\log \frac{\frac{p^{\epsilon}_{Y|Z}(y_0|x)}{p_Y(y_0)}}{\sum_{k=0}^{N}\frac{p^{\epsilon}_{Y|Z}(y_k|x)}{p_Y(y_k)}}\right] \tag{34}$$

$$= \mathop{\mathbb{E}}_{T}\left[\log\left(1 + \frac{p_Y(y_0)}{p^{\epsilon}_{Y|Z}(y_0|x)}\sum_{k=1}^{N}\frac{p^{\epsilon}_{Y|Z}(y_k|x)}{p_Y(y_k)}\right)\right] \tag{35}$$

$$\approx \mathop{\mathbb{E}}_{T}\left[\log\left(1 + \frac{p_Y(y_0)}{p^{\epsilon}_{Y|Z}(y_0|x)}N\mathop{\mathbb{E}}_{y\sim p_Y}\left[\frac{p^{\epsilon}_{Y|Z}(y|x)}{p_Y(y)}\right]\right)\right] \tag{36}$$

$$= \mathop{\mathbb{E}}_{T}\left[\log\left(1 + \frac{p_Y(y_0)}{p^{\epsilon}_{Y|Z}(y_0|x)}N\right)\right] \tag{37}$$

$$\geq \log N - \mathop{\mathbb{E}}_{(z,y)\sim p^{\epsilon}_{Z,Y}}\left[\log\frac{p^{\epsilon}_{Y|Z}(y|z)}{p_Y(y)}\right], \tag{38}$$

where N denotes the the number of negative samples.

We can rewrite the expectation in (38) into

$$\mathop{\mathbb{E}}_{(z,y)\sim p^{\epsilon}_{Z,Y}}\left[\log\frac{(1-\epsilon)p_{Y|Z}(y|z)+\epsilon p_Y(y)}{p_Y(y)}\right] \tag{39}$$

$$= \mathop{\mathbb{E}}_{(z,y)\sim p^{\epsilon}_{Z,Y}}\left[\log\frac{p_{Y|Z}(y|z)}{p_Y(y)}\right] + \mathop{\mathbb{E}}_{(z,y)\sim p^{\epsilon}_{Z,Y}}\left[\log\left(1-\epsilon+\epsilon\frac{p_Y(y)}{p_{Y|Z}(y|z)}\right)\right] \tag{40}$$

$$= (1-\epsilon)\mathop{\mathbb{E}}_{(z,y)\sim p_{Z,Y}}\left[\log\frac{p_{Y|Z}(y|z)}{p_Y(y)}\right] + \epsilon\mathop{\mathbb{E}}_{\substack{z\sim p_Z\\ y\sim p_Y}}\left[\log\frac{p_{Y|Z}(y|z)}{p_Y(y)}\right] \tag{41}$$

$$+ \mathop{\mathbb{E}}_{(z,y)\sim p^{\epsilon}_{Z,Y}}\left[\log\left(\frac{(1-\epsilon)p_{Y|Z}(y|z)+\epsilon p_Y(y)}{p_{Y|Z}(y|z)}\right)\right] \tag{42}$$

$$= (1-\epsilon)\left(I(Z;Y) + \mathop{\mathbb{E}}_{(z,y)\sim p_{Z,Y}}\left[\log\left(\frac{(1-\epsilon)p_{Y|Z}(y|z)+\epsilon p_Y(y)}{p_{Y|Z}(y|z)}\right)\right]\right) \tag{43}$$

$$+ \epsilon\left(\mathop{\mathbb{E}}_{\substack{z\sim p_Z\\ y\sim p_Y}}\left[\log\frac{p_{Y|Z}(y|z)}{p_Y(y)}\right] + \mathop{\mathbb{E}}_{\substack{z\sim p_Z\\ y\sim p_Y}}\left[\log\left(\frac{(1-\epsilon)p_{Y|Z}(y|z)+\epsilon p_Y(y)}{p_{Y|Z}(y|z)}\right)\right]\right) \tag{44}$$

$$= (1-\epsilon)\left(I(Z;Y) + \mathop{\mathbb{E}}_{(z,y)\sim p_{Z,Y}}\left[\log\frac{p^{\epsilon}_{Y|Z}(y|z)}{p_{Y|Z}(y|z)}\right]\right) + \epsilon\mathop{\mathbb{E}}_{\substack{z\sim p_Z\\ y\sim p_Y}}\left[\log\frac{p^{\epsilon}_{Y|Z}(y|z)}{p_Y(y)}\right] \tag{45}$$

$$= (1-\epsilon)\left(I(Z;Y) + H(Y|Z)\right) + \mathop{\mathbb{E}}_{(z,y)\sim p^{\epsilon}_{Z,Y}}\left[\log p^{\epsilon}_{Y|Z}(y|z)\right] + \epsilon H(Y) \tag{46}$$

$$= (1-\epsilon)\left(I(Z;Y) + H(Y|Z)\right) + \mathop{\mathbb{E}}_{(z,y)\sim p^{\epsilon}_{Z,Y}}\left[\log p^{\epsilon}_{Y|Z}(y|z)\right] \tag{47}$$

$$+ \epsilon\left(I(Z;Y) + H(Y|Z)\right) \tag{48}$$

$$= I(Z;Y) + H(Y|Z) + \mathop{\mathbb{E}}_{(z,y)\sim p^{\epsilon}_{Z,Y}}\left[\log p^{\epsilon}_{Y|Z}(y|z)\right]. \tag{49}$$

Table 2: Top-5/class average classification accuracy for various datasets. Comparison of four baseline loss functions to soft target InfoNCE.

	ImageNet	Tiny ImageNet	CIFAR-100	CellTypeGraph
NLL	94.95	93.84	98.37	80.57
Soft target cross-entropy	**96.46**	94.91	98.40	**81.41**
InfoNCE	95.43	93.67	98.40	80.00
Soft distribution InfoNCE	96.40	**95.06**	**98.78**	79.97
Soft target InfoNCE	96.43	94.91	98.51	81.27
	top-5 acc.	top-5 acc.	top-5 acc.	class-avg. acc.

To simplify notation we define

$$H_\epsilon(Y|Z) := \mathop{\mathbb{E}}_{(z,y)\sim p^\epsilon_{Z,Y}} \left[-\log p^\epsilon_{Y|Z}(y|z) \right]. \tag{50}$$

Inserting (49) back into (38), we obtain

$$I(Z;Y) + H(Y|Z) - H_\epsilon(Y|Z) \geq \log N - l^{\mathrm{SDInfoNCE}}(T). \tag{51}$$

Next we show that the l.h.s. of (51) is larger or equal to zero:

$$H_\epsilon(Y|Z) = H_\epsilon(Y,Z) - H(X) \tag{52}$$

$$\leq H(Y) + H(X) - H(X) = H(Y). \tag{53}$$

Since $H(Y) = I(Z;Y) + H(Y|Z)$, we have $I(Z;Y) + H(Y|Z) - H_\epsilon(Y|Z) \geq 0$.

A.4 Experiments

A.4.1 Classification accuracy

Table 2 shows top-5 classification accuracy for ImageNet, Tiny ImageNet, CIFAR-100 and class-average accuracy for CellTypeGraph for all baselines.

Here we provide further technical details for our classification experiments.

Image classification. Table 3 details the settings used across all baselines. As classification head we used a 3-layer MLP with hidden dimension 4096. On ImageNet, we used as base learning rate of 0.00055 for InfoNCE-type loss functions, and 0.0003 for cross-entropy-type baselines. On Tiny ImageNet, we used 0.00025 for InfoNCE-type baselines, and 0.0003 for cross-entropy-type baselines. On CIFAR-100, we used 0.0003 for InfoNCE, soft distribution InfoNCE and cross-entropy-type baselines, and 0.00055 for soft target InfoNCE. Note, we determined (base) learning rates based on a search over multiple samples from the interval $[0.00005, 0.003]$.

Node classification. Besides the parameter settings we described in the main experiments, all DeeperGCN models were trained with adjacency dropout (dropout prob. of 0.1), adding of edges (ratio of 0.1), node feature dropout and masking (prob. of 0.1), random normal noise (σ of 0.1) and random basis transforms.

A.4.2 Ablation experiments

Confidence calibration. Figure 6 shows the reliability diagram of the best performing ViT-B/16 models on Tiny ImageNet.

Table 3: Parameter settings used to train ViT-B/16 models across all baselines.

parameter	value
optimizer	AdamW
weight decay	0.05
layer-wise lr decay	0.65
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	100
label smoothing	0.1
MixUp	0.8
CutMix	1.0
augmentations	RandAug(9, 0.5)

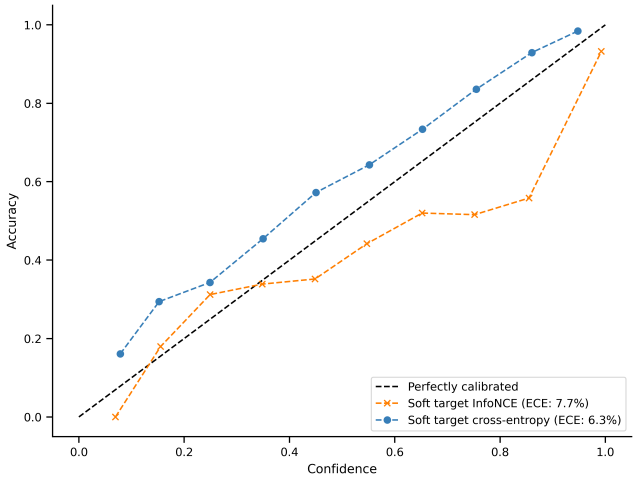


Figure 6: Reliability diagram of the best performing ViT-B/16 models on Tiny ImageNet, trained with soft target loss functions in combination with label smoothing, MixUp and CutMix.