Temperature as Uncertainty in Contrastive Learning

Oliver Zhang[1], **Mike Wu**[1], **Jasmine Bayrooti**[1], **Noah Goodman**[1,2]
Department of Computer Science[1] and Psychology[2]
Stanford University
Stanford, CA 94303
{ozhang, wumike, jbayrooti, ngoodman}@stanford.edu

Abstract

Contrastive learning has demonstrated great capability to learn representations without annotations, even outperforming supervised baselines. However, it still lacks important properties useful for real-world application, one of which is uncertainty. In this paper, we propose a simple way to generate uncertainty scores for many contrastive methods by re-purposing temperature, a mysterious hyperparameter used for scaling. By observing that temperature controls how sensitive the objective is to specific embedding locations, we aim to learn temperature as an input-dependent variable, treating it as a measure of embedding confidence. We call this approach "Temperature as Uncertainty", or TaU. Through experiments, we demonstrate that TaU is useful for out-of-distribution detection, while remaining competitive with benchmarks on linear evaluation. Moreover, we show that TaU can be learned on top of pretrained models, enabling uncertainty scores to be generated post-hoc with popular off-the-shelf models. In summary, TaU is a simple yet versatile method for generating uncertainties for contrastive learning. Open source code can be found at: https://github.com/mhw32/temperature-as-uncertainty-public.

1 Introduction

Representation learning through contrastive objectives has recently broken new ground, matching the performance of fully supervised methods on image classification [18, 15, 21, 5, 6, 12, 7, 40]. While contrastive learning has shown strong practical results, it still lacks some important properties for real-world decision-making. One such property is uncertainty, which plays an important role in recognizing and preventing errors. For example, uncertainty can be leveraged to find anomalies that are out-of-distribution (OOD), on which a model's predictions may be degraded or entirely out-of-place. However, current contrastive frameworks do not provide any indication of uncertainty as they learn one-to-one mappings from inputs to embeddings.

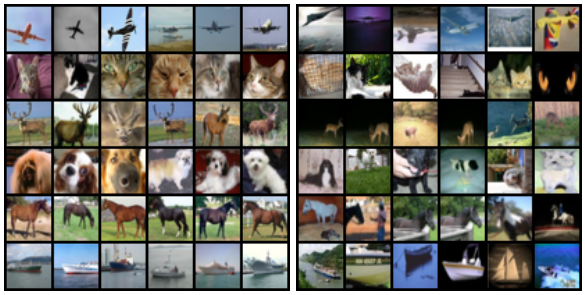

Figure 1: CIFAR10 Images on the left have high TaU certainty while images on the right have low TaU certainty.

Our work uses the temperature parameter to estimate the uncertainty of an input. While almost all contrastive frameworks include temperature in the objective, it historically has remained relatively unexplored compared to work on negative samples [35, 38], stop gradients [12, 7, 40], and transformation families [30, 33]. Recently, it has been shown that smaller temperature increases the model's

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

penalty on difficult negative examples [34]. With this intuition, we make temperature a learned, input-dependent variable. High temperature is tantamount to the model declaring that a training input is difficult. Temperature, therefore, can be viewed a form of uncertainty. We call this simple extension to the contrastive objective, "Temperature as Uncertainty" or TaU for short.

On benchmark image datasets, we show that TaU is useful for out-of-distribution detection, outperforming baseline methods for extracting uncertainty such as ensembling or Bayesian posteriors over weights. We also show that one can easily derive uncertainty on top of pretrained representations, making this approach widely applicable to existing model checkpoints and infrastructure.

2 Temperature as Uncertainty

To start, we give a brief overview of contrastive learning to motivate the approach. Suppose we have a dataset $\mathcal{D} = \{x_i\}_{i=1}^n$ of i.i.d image samples from $p(x)$, a distribution over a space of natural images X. Let $\mathcal{T}$ be some family of image transformations, $t : X \to X$, equipped with a distribution $p(t)$. The common family of visual transformations includes a random mix of cropping, color jitter, gaussian blurring, and horizontal flipping [37, 32, 42, 4, 15, 5].

Define an encoding function $f : X \to \mathbf{S}^{d-1}$ that maps an image to a L_2-normalized representation. Let f be parameterized by a deep neural network. The contrastive objective for the i-th example is:

$$\mathcal{L}(x_i) = \mathbf{E}_{t,t',t_{1:k}\sim p(t)} \mathbf{E}_{x_{1:k}\sim p(x)} \left[\log \frac{e^{f(t(x_i))\cdot f(t'(x_i))/\tau}}{\frac{1}{k}\sum_{j\in\{1:k\}} e^{f(t(x_i))\cdot f(t_j(x_j))/\tau}} \right] \tag{1}$$

where $x_{1:k} = \{x_1, \ldots, x_k\}$ represents k i.i.d. samples, and τ is the temperature. We call transformations of the same image "positive examples" and transformations of different images "negative examples". We chose to present Eq. 1 in a very general form based on the noise contrastive [24, 14] lower bound to mutual information [18, 26, 36, 33], although many popular frameworks like SimCLR [5] and MoCo-v2 [6] can be directly derived from Eq. 1.

Algorithm 1: TaU + SimCLR

```
# g: TaU encoder networks
# x: minibatch of images
# scale: constant (e.g. 0.1)

# encode augmentation and return
# embedding and temperature
# inv_tau is used for stability
emb1,tau1=g(random_aug(x))
emb2,tau2=g(random_aug(x))
emb1=norm(emb1)  # L2 norm
emb2=norm(emb2)
emb=cat(emb1,emb2)
# rescale for numerical stability
tau1 = sigmoid(tau1) / scale
tau2 = sigmoid(tau2) / scale
tau=cat(tau1,tau2)
# compute similarities
pos_dps = sum(emb1*emb2) * tau1
neg_dps = emb@emb.T * tau
# mask out identity comparisons
neg_dps = simclr_mask(neg_dps)
# parametric log softmax
loss=-pos_dps+logsumexp(neg_dps)
```

Since Eq. 1 uses a dot product as a distance function, the role of temperature, τ is to scale the sensitivity of the loss function [34]. A τ closer to 0 would accentuate when representations are different, resulting in larger gradients. In the same vein, a larger τ would be more forgiving of such differences. In practice, varying τ has a dramatic impact on embedding quality. Traditionally, there are fixed values for τ that the authors of a contrastive framework have painstakingly tuned.

We decide to learn an input-dependent temperature. In accordance with previous observations, learning an input-dependent temperature would amount to an embedding sensitivity for every input. In other words, *a measure of representation uncertainty*. Inputs with high temperature suggest more uncertainty as the objective is more invariant to displacements, whereas inputs with low temperature suggest less uncertainty as the objective is more sensitive to changes in embedding location.

Implementing this idea is very straightforward. We can replace τ in Eq. 1 with $\tau(t(x_i))$, overloading notation to define a mapping $\tau : X \to (M, \infty)$ for some lower bound M. We call this new objective TaU, or **T**emperature **a**s **U**ncertainty. In practice, we edit the encoder network $f(x)$ to return $d + 1$ entries, the first d of which are the embedding of x, and the last entry being the uncertainty for x. A sigmoid is used to bound τ to be positive and a fixed constant is used to set a lower bound M for stability. See Algo. 1 for pseudo-code.

3 Related Work

Learned Temperature Methods which learn temperature can be found in supervised learning [41, 3], model calibration [13, 22], language supervision [27], and few-shot learning [25, 28]. In most

of these approaches, temperature is treated as a global parameter when it is learned, not as a function of the input as in TaU. The only example, to the best of our knowledge, of learned temperature as a function of the input is [22], which uses temperature for calibration. Instead, we use temperature in the context of self-supervised learning and apply it to OOD detection.

Uncertainty in Deep Learning There is a rich body of work in adding uncertainty to deep learning models [10, 2], of which we highlight a few. Most straightforward is ensembling of neural networks [31], where multiple copies are trained with different parameter initializations to find many local minima. Further work attempts to enforce ensemble diversity for more variance [20, 1]. Another popular approach of uncertainty is through Bayesian neural networks [11], of which the most practical formulation is Monte Carlo dropout [9]. This approach frames using dropout layers during training and test time as equivalent to sampling weights from a posterior distribution over model parameters. Finally, most relevant is "hedged instance embeddings" [23], which edits the contrastive encoder f to map an image to a Gaussian distribution, rather than a point embedding. The primary drawbacks of this approach are (1) computational cost as it requires multiple samples, and (2) it is not proven to work in high dimensions. In our experiments, we compare these baselines to TaU.

OOD Detection Existing OOD algorithms mostly derive outlier scores on top of predictions made by large supervised neural networks trained on the inlier dataset, such as using the maximum softmax probability [17], sensitivity to parameter perturbations [19], or Gram matrices on activation maps [29] as the outlier score. While these methods work very well, reaching near ceiling performance, they require human annotations, which may not be available.

4 Experiments

A primary application of uncertainty is to find abnormal or anomalous inputs. We aim to show that using TaU temperatures as uncertainty is effective for out-of-distribution (OOD) detection [17, 19, 29], with the added bonus of sacrificing little to no performance on downstream tasks.

OOD Detection We study OOD detection, where inputs from an anomalous distribution are fed to a trained model. A well-performing metric should assign high uncertainty to these OOD inputs, thereby making it possible to classify whether an input is OOD. The specific threshold is chosen by observing the input distribution (e.g., by picking the confidence which has a 5% true positive rate).

We train TaU on CIFAR10 and consider three different OOD sets: CIFAR100, SVHN, and TinyImageNet. We note that CIFAR10 and CIFAR100 are very similar, whereas SVHN is the most dissimilar. To measure performance, we compute AUROC on correctly classifying an example as OOD. We compare TaU to several baselines. Assuming unrestricted computational power and memory, one could fit k-nearest neighbors on the entire training corpus, and treat the average distance of a new example to k neighbors as an uncertainty score. We try out $k = [1, 3, 10, 32, 100]$ with $k = 10$ working the best. For other baselines, please refer to Sec. 3. For MC Dropout, we take the average of the standard deviations of each dimension of the embedding vectors. This is to find the uncertainty of the embedding; the original implementation finds the uncertainty in the final predictions.

Table 1: **Downstream Out-of-Distribution Detection**: comparison of TaU to several popular baselines for uncertainty on deep neural networks. Out-of-distribution AUROC is reported.

Method (CIFAR10)	**CIFAR100**	**SVHN**	**TinyImageNet**
TaU + SimCLR	**0.746**	0.964	**0.760**
TaU + MoCo-v2	0.728	**0.968**	0.746
SimCLR + kNN	**0.746**	0.829	0.756
MoCo-v2 + kNN	0.712	0.800	0.726
SimCLR + MC Dropout [9]	0.504	0.684	0.512
Supervised + MC Dropout [9]	0.659	0.745	0.722
Hedged Instance Embedding [23]	0.509	0.834	0.508
Ensemble of 5 SimCLRs	0.532	0.525	0.513

From Table 1, we observe that on CIFAR100 and TinyImageNet – two image corpora with similar content as CIFAR10 – TaU outperforms (or matches) all baselines, though only surpassing SimCLR + kNN by a small margin of 0-2%. However, for SVHN – an image corpus very different in content to CIFAR10 – TaU outperforms all baselines by at least 13%. In fact, we find most baselines do not

generalize well to contrastive learning, as many perform near chance AUROC. Even prior methods specifically for contrastive uncertainty [23] do not consistently perform well. The exception is using kNN *with the caveat* that the OOD set was not too far from the training set. Nearest neighbors fundamentally relies on a good distance function, which is achievable when the OOD input can be properly embedded. But in cases when we are truly OOD, it may not be clear where to embed an anomalous image. In these cases, as with SVHN, kNN approaches struggle.

Linear Evaluation Although we have shown that TaU uncertainties can detect OOD inputs, they would not be much good if it came at a large cost of performance on downstream tasks. To show this is not the case, we measure performance through both linear evaluation [5] and k-nearest neighbors on the training set [42] (where the predicted label for a test example is the label of the closest example in the training set). Please refer to the appendix for experiment details.

Table 2: **Downstream Image Classification**: mean and standard deviation in accuracy are measured over three runs with different random seeds. The best performing models are bolded.

Method	kNN Eval	Linear Eval
TaU + SimCLR	0.762 ± 0.001	0.750 ± 0.003
TaU + MoCo-v2	0.709 ± 0.004	0.690 ± 0.004
SimCLR	$\mathbf{0.787} \pm 0.004$	$\mathbf{0.775} \pm 0.002$
MoCo-v2	0.734 ± 0.004	0.720 ± 0.005

From Table 2, we observe TaU to perform only slightly worse than their deterministic counterparts, with a small reduction of 2-3 percentage points on both linear and k-nearest neighbors evaluation. While there is a non-zero cost to adding uncertainty, we believe that the trade is worthwhile.

Uncertainty on Pretrained Models We next show that TaU can be finetuned on top of pretrained models, enabling uncertainties to be generated post-hoc on popular off-the-shelf checkpoints. Specifically, we finetune on supervised, SimCLR, BYOL, and CLIP embeddings. All models were pretrained using ResNet50 on ImageNet with the exception of CLIP, which uses a ViT [8]. We finetune TaU uncertainties for 40 epochs, and all images were reshaped to 224 by 224 pixels.

Table 3: **Out-of-Distribution Detection using Pretrained Embeddings**: using TaU to generate uncertainties for several pretrained models. Out-of-distribution AUROC is reported.

Method (ImageNet)	CIFAR10	CIFAR100	SVHN	TinyImgNet	LSUN	COCO	CelebA
TaU + Supervised	**0.913**	**0.874**	**0.978**	**0.771**	0.657	0.458	0.657
TaU + SimCLR [5]	0.823	0.870	0.968	0.747	0.552	0.554	0.717
TaU + BYOL [12]	0.763	0.808	0.955	0.686	0.471	0.497	0.840
TaU + CLIP [27]	0.056	0.044	0.071	0.154	**0.779**	**0.579**	**0.883**

Table 3 reports AUROC for OOD detection for a wide survey of outlier datasets. We find that for supervised, SimCLR, and BYOL embeddings, the learned TaU uncertainties are largely able to classify OOD inputs. The exception is with COCO, likely due to a close similarity with ImageNet data points. However, CLIP surprisingly faces the opposite problem with low OOD scores for most datasets but outperforming in COCO and LSUN. Further work could explore whether CLIP's behavior is due to differences in objective, architecture, or training.

5 Limitations and Future Work

We presented TaU, a simple method for adding uncertainty into contrastive learning objectives by repurposing temperature as uncertainty. In our experiments, we compared TaU to existing benchmark algorithms and found competitive downstream performance, in addition to TaU uncertainties being useful for out-of-distribution detection. We then demonstrated how uncertainty can be added to already trained model checkpoints, enabling practitioners to reuse computation.

We discuss an important limitation: our approach is restricted to contrastive algorithms built on NCE. Other approaches, such as SimSiam [7], BYOL [12], and Barlow Twins [40], replace negative examples entirely with stop gradients, where we find limited success with TaU. Future work can also explore TaU-like techniques for detecting corrupted or adversarial examples.

References

[1] Mahdieh Abbasi, Arezoo Rajabi, Christian Gagné, and Rakesh B Bobba. Toward adversarial robustness by diversity in an ensemble of specialized deep neural networks. In *Canadian Conference on Artificial Intelligence*, pages 1–14. Springer, 2020.

[2] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 2021.

[3] Atish Agarwala, Jeffrey Pennington, Yann Dauphin, and Sam Schoenholz. Temperature check: theory and practice for training models with softmax-cross-entropy losses. *arXiv preprint arXiv:2010.07344*, 2020.

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[5] 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.

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[8] 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. In *International Conference on Learning Representations*, 2020.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[10] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*, 2021.

[11] Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. In *Case Studies in Applied Bayesian Data Science*, pages 45–87. Springer, 2020.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[18] 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.

[19] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[20] Ling Liu, Wenqi Wei, Ka-Ho Chow, Margaret Loper, Emre Gursoy, Stacey Truex, and Yanzhao Wu. Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness. In *2019 IEEE 16th international conference on mobile ad hoc and sensor systems (MASS)*, pages 274–282. IEEE, 2019.

[21] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[22] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection. 2018.

[23] Seong Joon Oh, Kevin Murphy, Jiyan Pan, Joseph Roth, Florian Schroff, and Andrew Gallagher. Modeling uncertainty with hedged instance embedding. *arXiv preprint arXiv:1810.00319*, 2018.

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

[25] Boris N Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *arXiv preprint arXiv:1805.10123*, 2018.

[26] 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.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[28] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Self-supervised knowledge distillation for few-shot learning. *arXiv preprint arXiv:2006.09785*, 2020.

[29] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with in-distribution examples and gram matrices. *arXiv preprint arXiv:1912.12510*, 2019.

[30] Alex Tamkin, Mike Wu, and Noah Goodman. Viewmaker networks: Learning views for unsupervised representation learning. *arXiv preprint arXiv:2010.07432*, 2020.

[31] Sean Tao. Deep neural network ensembles. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 1–12. Springer, 2019.

[32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer, 2020.

[33] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

[34] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. *arXiv preprint arXiv:2012.09740v2*, 2021.

[35] Mike Wu, Milan Mosse, Chengxu Zhuang, Daniel Yamins, and Noah Goodman. Conditional negative sampling for contrastive learning of visual representations. *arXiv preprint arXiv:2010.02037*, 2020.

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

[37] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[38] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Delving into inter-image invariance for unsupervised visual representations. *arXiv preprint arXiv:2008.11702*, 2020.

[39] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[40] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

[41] Xu Zhang, Felix Xinnan Yu, Svebor Karaman, Wei Zhang, and Shih-Fu Chang. Heated-up softmax embedding. *arXiv preprint arXiv:1809.04157*, 2018.

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

A Training Hyperparameters

Pretraining For all models, we use a representation dimensionality of 128. We use the LARS optimizer [39] with learning rate 1e-4, weight decay 1e-6, batch size 128 for 200 epochs, as described in [5]. For baseline models (no uncertainty), we use a fixed temperature $\tau = 0.1$. For MoCO-V2, we use $K = 65536$ negative samples with a momentum of 0.999 for updating the memory queue. We use the same optimizer as described above but with learning rate 1e-3. For CIFAR10, all images are resized to 32x32 pixels; For ImageNet, all images are resized to 256x256 pixels (with 224x224 cropping size). During pretraining, we use random resized crop, color jitter, random grayscale, random gaussian blur, horizontal flipping, and pixel normalization (with ImageNet statistics). During testing, we only center crop and do pixel normalization. For encoders we train from scratch, we use ResNet18 [16] for encoders f. We adapted the Pytorch Lightning Bolts implementations of SimCLR and MoCo-v2, found here: `https://github.com/PyTorchLightning/lightning-bolts`.

For the larger models, we downloaded existing SimCLR (ResNet50) checkpoints trained on ImageNet from `https://github.com/google-research/simclr` and converted it to PyTorch checkpoints using `https://github.com/Separius/SimCLRv2-Pytorch`.

Downstream Classification We freeze encoder parameters, remove the final L_2 normalization, and append a 2-layer MLP with hidden dimension of 128 and ReLU nonlinearity. For optimization, we use SGD with batch size 256, learning rate 1e-4, weight decay 1e-6, and cosine learning rate schedule that drops at epoch 60 and 80, with a total of 100 epochs.

Optimization Stability When optimizing the TaU objective, we found optimization instability where $\tau(z)$ would either collapse to 0 or converge to $\pm\inf$ if left unbounded. We found it crucial to employ some tricks for optimization stability. First, we follow Neumann et al. and have our network predict some $\alpha(z) = \frac{1}{\tau(z)}$ instead of $\tau(z)$ directly [22]. This changes the training dynamics but does not change the underlying equation. Second, we bound $\alpha(z)$ to between 0 and 1 using a sigmoid function. Finally, we divide $\alpha(z)$ by 0.1, which helps initialize the temperature to be in the same range as fixed-temperature models. When using the uncertainty for out-of-distribution detection, we found that using the pre-sigmoid $\alpha(z)$ worked much better than the post-sigmoid $\alpha(z)$, as the differences between post-sigmoid values became indistinguishable using float32.