UNCERTAINTY-AWARE SELF-SUPERVISED LEARNING WITH INDEPENDENT SUB-NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning methods are state-of-the-art across a wide range of tasks in computer vision, natural language processing, and multimodal analysis. Estimating the epistemic – or model – uncertainty of self-supervised model predictions is critical for building trustworthy machine learning systems in crucial applications, such as medical diagnosis and autonomous driving. Ensembling a neural network is an effective approach to estimating uncertainty. However, deep ensembles induce high computational costs and memory demand. This is even more challenging in the case of self-supervised deep learning, where even a single network is computationally demanding. Moreover, most existing model uncertainty techniques are built for supervised deep learning. Motivated by this, we propose a novel approach to making self-supervised learning probabilistic. We introduce an uncertainty-aware training regime for self-supervised models with an ensemble of independent sub-networks and a novel loss function for encouraging diversity. Our method builds a sub-model ensemble with high diversity – and consequently, well-calibrated estimates of model uncertainty – at low computational overhead over a single model, while performing on par with deep self-supervised ensembles. Extensive experiments across different tasks, such as in-distribution generalization, out-of-distribution detection, dataset corruption, and semi-supervised settings, demonstrate that our approach increases prediction reliability. We show that our method achieves both excellent accuracy and calibration, improving over existing ensemble methods in a wide range of self-supervised architectures for computer vision, natural language processing, and genomics data.

1 INTRODUCTION

Estimating uncertainty is an important component for developing reliable and trustworthy machine learning algorithms used to make predictions for important decisions, such as medical diagnosis (Azizi et al., 2021), drug discovery (Jiménez-Luna et al., 2020), and autonomous driving (Kaya et al., 2022; Can et al., 2022). In recent years, self-supervised learning methods have achieved cutting-edge performance across a wide range of tasks in natural language processing (NLP; (Devlin et al., 2018; Brown et al., 2020), computer vision (Chen et al., 2020; Bardes et al., 2021; Grill et al., 2020), multimodal learning (Radford et al., 2021; Li et al., 2022; Shi et al., 2022), and bioinformatics (Gündüz et al., 2021). In contrast to supervised techniques, these models learn representations of the data without relying on costly human annotation.

Despite remarkable progress in recent years, self-supervised models do not allow practitioners to inspect the model's confidence. This problem is non-trivial, given the degree to which critical applications rely on self-supervised methods. Therefore, quantifying the predictive uncertainty of self-supervised models is key to more reliable downstream tasks. Here, we follow the definition of reliability as described by Plex (Tran et al., 2022), in which the ability of a model to work consistently across many tasks is assessed. In particular, Tran et al. (2022) introduces three general desiderata of reliable machine learning systems: a model should generalize robustly to *new tasks*, as well as *new datasets*, and represent the associated *uncertainty* in a faithful manner.

Predictive uncertainty is classified as epistemic (model) and aleatoric (data) uncertainty. Aleatoric uncertainty is inherent to the data and arises from natural stochasticity in observations, for instance, due to class overlap, label noise, or multi-label disagreement. On the other hand, epistemic uncer-

tainty concerns the learning hypothesis parameterized through model weights. It is caused by limited data or knowledge and is therefore reducible. Addressing this source of uncertainty is crucial in order to build methods that generalize robustly to unseen data and avoid wrong, overconfident predictions. Importantly, evaluating the quality of predictive uncertainties is challenging, as the 'ground-truth' uncertainty is usually not available.

Bayesian neural networks (BNNs; Neal (2012)) and neural ensemble networks (Hansen & Salamon, 1990) are the common approaches for capturing distributions over parameters of neural networks. While the Bayesian paradigm, with its principled approach to inference, is the canonical answer to the problem of uncertainty quantification, it is not well-suited to self-supervised methods. Bayesian deep learning, which usually requires samples from the posterior distribution, scales poorly to the large architectures that prevail in this field and relies fundamentally on the availability of true labels y for maximum-a-posterior (MAP) estimation. During pretext-task learning (pretraining) in self-supervised and unsupervised problems, we do not have access to such label information. This prevents us from taking advantage of BNNs during the pretraining of self-supervised models.

On the other hand, deep ensembles (Lakshminarayanan et al., 2017) have been proposed as a heuristic alternative to Bayesian learning. By adding additional parameter instances through training multiple networks in parallel, deep ensembles make learning probabilistic. Despite their lack of theoretical foundation, deep ensembles have been successful in practice and become gold standard in estimating uncertainty of neural networks. The high performance of deep ensembles, however, comes at the expense of drastically increased costs. Subsequent work, such as BatchEnsemble (Wen et al., 2020), Masksemble (Durasov et al., 2021), MIMO (Havasi et al., 2021), and FiLM-Ensemble (Ozgur Turkoglu et al., 2022) has attempted to address this problem by creating ensembles more efficiently. While these models succeed at speeding up training and/or inference, they exhibit a substantial performance gap compared to the deep ensemble, lacking both prediction quality and calibration.

In this work, we introduce a simple and scalable framework for incorporating predictive uncertainty estimates into the self-supervised learning approach while preserving performance with a negligible increase in computational cost. Our contributions can be summarized as follows:

- We propose a novel way to extend self-supervised learning to be probabilistic, which enables self-supervised models to provide uncertainty information for predictions.
- We develop a complementary uncertainty-aware loss function to enforce diversity among the independent sub-networks.
- We perform extensive empirical analyses to highlight the benefits of learning probabilistic representations. We demonstrate that this inexpensive probabilistic modification achieves very competitive (in most cases, better) predictive performance: 1) on in-distribution (IND) and out-of-distribution (OOD) tasks; 2) in semi-supervised settings; 3) learns a better predictive performance-uncertainty trade-off than compared baselines (i.e., exhibits high predictive performance and low uncertainty on IND datasets as well as high predictive performance and high uncertainty on OOD datasets).

2 Related work

Self-supervised learning For most large-scale modeling problems, learning under full supervision is severely inhibited by the scarcity of annotated samples. Self-supervised learning techniques, which solve *pretext tasks* (Devlin et al., 2018) to generate labels from (typically abundant) unlabeled data, have proven to be a powerful remedy to this bottleneck. The learned feature maps can serve as a starting point for *downstream* supervised tasks, such as classification, object detection or sentiment analysis, with a substantially reduced need for labeled examples (Jaiswal et al., 2020). Alternatively, the downstream application may directly use the extracted representation for problems such as anomaly OOD detection. Self-supervision has been shown to enhance robustness against numerous sources of corruption and to even improve on purely supervised approaches in certain tasks (Hendrycks & Dietterich, 2019). However, its success depends on the quality of pretraining, for which (Tran et al., 2022) recently proposed a number of desiderata. While there have been attempts to make pretraining more robust by preventing embedding collapse (Bardes et al., 2021) or boosting performance in OOD detection (Winkens et al., 2020; Sehwag et al., 2021), the aspect

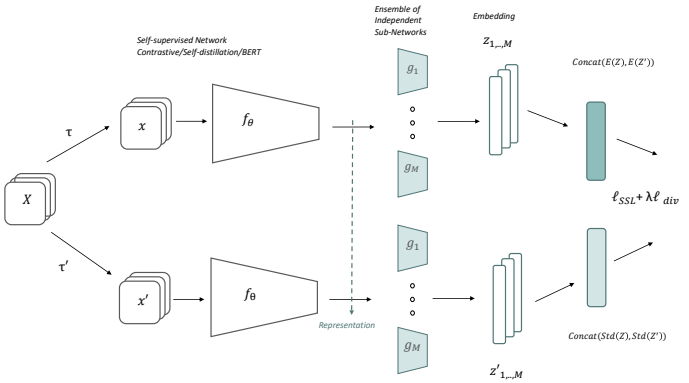


Figure 1: Illustration of our proposed method. Given a batch $\boldsymbol{X}$ of input samples, two different views $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$ are produced for each sample, which are then encoded into representations by the encoder network $f_{\boldsymbol{\theta}}$. The representations are projected to the ensemble of independent sub-networks g_m, where each sub-network produces embedding vectors $\boldsymbol{z}$ and $\boldsymbol{z}'$. The mean value of these embeddings are passed to the self-supervised loss, while their standard deviation is used for the diversity loss. Finally, the total loss is computed by combination of the two loss components.

of *uncertainty-awareness* has been studied to a lesser extent in the self-supervised context. Motivated by this, we present a simple way to make self-supervised learning uncertainty-aware during pretext-task learning.

Model uncertainty estimation and uncertainty-awareness Uncertainty can be viewed as a state of missing information in which it is impossible to exactly describe an outcome of interest (Ghahramani, 2015). In predictive modeling, awareness of such uncertainty is obviously desirable. Uncertainty estimation helps practitioners assess the trustworthiness of model outcomes, thereby permitting suitable interventions when the model is likely to be wrong. Bayesian learning handles uncertainty in a natural manner but is not applicable to self-supervised learning in the absence of labeled data. For example, SNGP (Liu et al., 2020) creates uncertainty-awareness via the preservation of input-space distances. Subsequent studies, including modifications using Mahalanobis instead of Euclidean distances (Vazhentsev et al., 2022), suggest that SNGP works well in representation learning for computer vision and NLP tasks. Crucially, however, the underlying single-model approach is inherently unable to capture the epistemic part of uncertainty. The stochastic embeddings learned in (Park et al., 2022) resolve this but depend on the use of contrastive learning, as does loss-temperature scaling (Zhang et al., 2021), and a number of Gaussianity assumptions. (Hu & Khan, 2021) promote a more holistic approach rooted in evidential learning yet rely on additional datasets from outside the training manifold. Sidestepping such shortcomings, we propose to employ the well-established ensembling approach to enforce uncertainty-awareness in a simple and effective manner.

Ensemble learning Deep Ensembles (Lakshminarayanan et al., 2017) comprise a set of M neural networks that independently train on the same data using random initialization. Deep ensembles often outperform other approaches in terms of calibration and predictive accuracy (Ovadia et al., 2019; Gustafsson et al., 2020; Ashukha et al., 2020), but their naive application incurs high computational complexity, as training, memory, and inference cost multiplies with the number of base learners. BatchEnsemble (Wen et al., 2020) introduces multiple low-rank matrices with little training and storage demand, whose Hadamard products with a shared global weight matrix mimic an ensemble of models. FiLM-Ensemble (Ozgur Turkoglu et al., 2022) uses feature-wise linear modulation to construct an ensemble for uncertainty estimation. Masksensemble (Durasov et al., 2021) builds upon Monte Carlo dropout (Gal & Ghahramani, 2016) and proposes a learnable (rather than a random) selection of masks used to drop certain network neurons. MIMO (Havasi et al., 2021) uses ensembles of sub-networks diverging only at the beginning and end of the parent architecture – thus sharing the vast majority of weights – in order to obtain multiple predictions with a single forward pass. At test time, several copies of each sample are fed to the enlarged input layer, and the multi-head last layer returns an according number of predictions. Although these methods reduce the inference time and computational resources required at training, the benefits are limited to larger pretraining model that is used in self-supervised learning.

3 METHOD

We propose a simple principle to: 1) estimate uncertainty during pretraining of self-supervised deep learning, and 2) improve predictive uncertainty. As depicted in Figure. 1, our proposed method can be readily applied to the most recent trends in self-supervised learning (Caron et al., 2021; Grill et al., 2020; Chen et al., 2020; Devlin et al., 2019; Gündüz et al., 2021; Klein & Nabi, 2022) and is based on a joint embedding architecture. In the following sections, we first describe our proposed uncertainty-aware self-supervised model, followed by the diversity loss, and then discuss ensemble diversity and computational cost.

Uncertainty-aware self-supervised learning via independent sub-networks

Given a randomly sampled mini-batch of data $\boldsymbol{X} = \{\boldsymbol{x}_k\}_{k=1}^N$, $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^p$, the transformer function derives two augmented views $\tilde{\boldsymbol{x}} = \tau(\boldsymbol{x}), \tilde{\boldsymbol{x}}' = \tau'(\boldsymbol{x})$ for each sample in $\boldsymbol{X}$. The augmented views are obtained by sampling τ, τ' from a distribution over suitable data augmentations, such as masking parts of sequences (Baevski et al., 2022; Devlin et al., 2019), partially masking image patches (He et al., 2022), or applying image augmentation techniques (Chen et al., 2020).

The two augmented views $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$ are then fed to an encoder network $f_{\boldsymbol{\theta}}$ with trainable parameters $\boldsymbol{\theta} \subseteq \mathbb{R}^d$. The encoder (e.g., ResNet-50 (He et al., 2016), ViT (Dosovitskiy et al., 2020)) maps distorted samples to a set of corresponding features. We call the output of the encoder the *representation*. Afterwards, the representation features are transformed by M independent sub-networks $\{g_{\boldsymbol{\phi_m}}\}_{m=1}^M$ with trainable parameters $\boldsymbol{\phi}_m$. The ensemble constructs from the representation M different q-dimensional *embedding* vectors $\{\boldsymbol{z}_m\}_{m=1}^M$, $\{\boldsymbol{z}'_m\}_{m=1}^M$, respectively, for $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{x}}'$. We modify the conventional self-supervised loss and replace the usual $\boldsymbol{z}_m, \boldsymbol{z}'_m$ with the mean values $\bar{\boldsymbol{z}}, \bar{\boldsymbol{z}}'$ over $m = 1, \ldots, M$. Averaging over the embeddings generated by the M sub-networks improves predictive performance. For example, in the case of contrastive learning (Chen et al., 2020), the self-supervised loss ℓ_{ssl} with temperature $t > 0$ and cosine similarity $\text{sim}(\cdot, \cdot)$ is computed as:

$$\ell_{\text{ssl}}\left(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}'_k\right) = -\log \frac{\exp(\text{sim}(\bar{\boldsymbol{z}}_k, \bar{\boldsymbol{z}}'_k)/t)}{\sum_{i=1}^{2N} \mathbb{I}_{[k \neq i]} \exp(\text{sim}(\bar{\boldsymbol{z}}_k, \bar{\boldsymbol{z}}_i)/t)}. \tag{1}$$

Diversity is a key component of successful model ensembles (Fort et al., 2019). To this end, we design a new loss function for encouraging diversity during the training of the sub-networks. We define the diversity regularization term ℓ_{div} as a hinge loss over the difference of the standard deviation across the embedding vectors $\{\boldsymbol{z}_{k,m}\}_{m=1}^M$, $\{\boldsymbol{z}'_{k,m}\}_{m=1}^M$ to a minimum diversity of $\alpha > 0$. The standard deviation is the square root of the element-wise variance $\{\sigma^2_{k,o}\}_{o=1}^q$:

$$\sigma^2_{k,o} = \tfrac{1}{M-1} \textstyle\sum_{m=1}^M (z_{k,m,o} - \bar{z}_{k,o})^2 + \epsilon, \tag{2}$$

where $\epsilon > 0$ is a small scalar preventing numerical instabilities. The diversity regularization function is then given by:

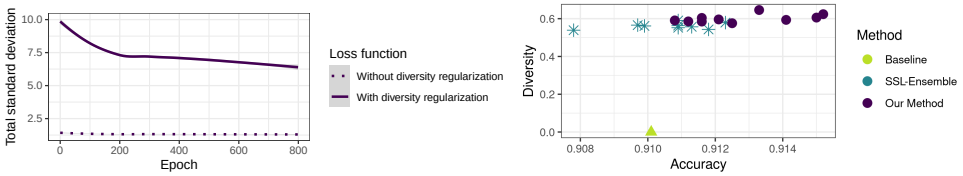
$$\ell_{\text{div}}\left(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}'_k\right) = \textstyle\sum_{o=1}^q \max\left(0, \alpha - \sigma_{k,o}\right) + \max(0, \alpha - \sigma'_{k,o}). \tag{3}$$

The objective of the diversity loss is to encourage disagreement among sub-networks by enforcing the element-wise standard deviations to be close to $\alpha > 0$ and to thus prevent the embeddings from collapsing to the same vector. Figure 2a motivates the importance of the diversity loss on the total sum of standard deviation between different sub-networks, which increases by adding the diversity loss. The total loss is calculated by combining the self-supervised loss (Eq. 1) and the diversity loss (Eq. 3), where the degree of regularization is controlled by a tunable hyperparameter $\lambda \geq 0$:

$$\ell\left(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}'_k\right) = \ell_{\text{ssl}}\left(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}'_k\right) + \lambda \cdot \ell_{\text{div}}\left(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}'_k\right). \tag{4}$$

Finally, the total loss is aggregated over all the pairs in minibatch $\boldsymbol{X}$:

$$\mathcal{L}_{\text{total}} = \tfrac{1}{N} \textstyle\sum_{k=1}^N \ell\left(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}'_k\right). \tag{5}$$



(a) (b)

Figure 2: (a) **Total standard deviation**: sum of all standard deviations between independent sub-networks during training. Training with diversity loss (Eq. 3) increases the standard deviation and improves the diversity between independent sub-networks. (b) **Diversity analysis**: prediction diversity disagreement vs. achieved accuracy on CIFAR-10. Our method is on par with the deep self-supervised ensemble in terms of both accuracy and diversity disagreement. Models in the top right corner are better.

Diversity analysis Diversity is an important feature for powerful model ensembles and reflects the degree of independence among its members (Zhang & Ma, 2012). We follow (Fort et al., 2019) to quantify the diversities among the ensemble of sub-networks. Specifically, we report the diversities in terms of *disagreement score* between the members' predictive distributions. Diversity disagreement is defined as the fraction of *distance disagreement* divided by *1 - accuracy*, where the distance disagreement between two classification models h_i and h_j is calculated as $\frac{1}{N}\sum_{k=1}^{N}\left[h_i(\mathbf{x}_k) \neq h_j(\mathbf{x}_k)\right]$, with N denoting the number of samples. Figure 2b compares the diversity disagreement between our method, a deep ensemble, and the single-network baseline. The results clearly indicate that our proposed method achieves comparable results with deep self-supervised ensembles in terms of both accuracy and diversity disagreement.

Computational cost We analyze the efficiency of our proposed method in terms of the resources needed for training. Table. 1 compares the relative cost of our proposed method with the self-supervised baseline and ensembles in terms of the number of parameters, memory demand, and computation time. A more detailed description of relative cost is provided in Figure. 6 Appendix A.

Table 1: Computational cost in 4 DGX-A100 40G GPUs (PyTorch) on CIFAR 10.

Method	Members	Parameters(M)	Memory / GPU	Time / 800-ep.
Baseline (SSL)	1	28	9 G	3.6 (h)
SSL-Ensemble	3	3×28	3×9 G	3× 3.6 (h)
SSL-Ensemble	10	10×28	10×9 G	10×3.6 (h)
Our method	3	37	9.2 G	3.6 (h)
Our method	10	68.1	10 G	3.8 (h)

4 Experimental setup

We perform several experiments with a variety of self-supervised methods to improve reliability during both pretext-task learning and downstream tasks (fine-tuning). The source code and our models are provided in `https://anonymous.4open.science/r/Uncertainty_aware_SSL-95D7/README.md`

Deep self-supervised network architecture Our proposed approach builds on two recent popular self-supervised models in computer vision: i) **SimCLR** (Chen et al., 2020) is a contrastive learning framework that learns representations by maximizing agreement on two different augmentations of the same image, employing a contrastive loss in the latent embedding space of a convolutional network architecture (e.g., ResNet-50 (He et al., 2016)), and ii) **DINO** (Caron et al., 2021) is a self-distillation framework in which a student vision transformer (ViT; Dosovitskiy et al. (2021)) learns to predict global features from local image patches supervised by the cross-entropy loss from a momentum teacher ViT's embeddings. Furthermore, we study the impact of our approach in NLP and modify **SCD** (Klein & Nabi, 2022), which applies the bidirectional training of transformers to language modeling. Here, the objective is self-supervised contrastive divergence loss. Lastly, we implement a probabilistic version of **Self-GenomeNet** (Gündüz et al., 2021), a contrastive self-

supervised learning algorithm for learning representations of genome sequences. More detailed descriptions of the employed configurations are provided in Appendix A.3.

Deep independent sub-networks We implement M independent sub-networks on top of the encoder, for which many possible architectures are conceivable. For our experiments on computer vision datasets, we consider a convex architecture where each network includes a multi-layer perceptron (MLP) with two layers of 2048 and 128 neurons, respectively, with ReLU as a non-linearity and followed by batch normalization (Ioffe, 2017). Each sub-network has its own independent set of weights. For the NLP dataset, the projector MLP contains three layers of 4096 neurons each, also using ReLU activation's as well as batch normalization. For the genomics dataset, our ensemble of sub-networks includes one fully connected layer with an embedding size of 256.

Optimization For all experiments on image datasets based on DINO and SimCLR, we follow the suggested hyperparameters and configurations by the paper (Caron et al., 2021; Chen et al., 2020). Implementation details for pretraining with DINO on the 1000-classes ImagetNet dataset without labels are as follows: coefficients ϵ, α, and λ are respectively set to $0.0001, 0.15$, and 2 in Eq.2, 3, and 4. We provide more details in ablation studies (Section 6) on the number of sub-networks and the coefficients λ and α used in the loss function. The encoder network $f_{\boldsymbol{\theta}}$ is either a ResNet-50 (He et al., 2016) with 2048 output units when the baseline is SimCLR (Chen et al., 2020) or ViT-s (Dosovitskiy et al., 2020) with 384 output units when the baseline is DINO (Caron et al., 2021). The best prediction and calibration performance is achieved when the number of sub-networks is 5. The training protocol follows the suggested settings by Caron et al. (2021).

Datasets We use the following datasets in our experiments: **CIFAR-10/100** (Krizhevsky, 2009) are subsets of the tiny images dataset. Both datasets include 50,000 images for training and 10,000 validation images of size 32×32 with 10 and 100 classes, respectively. **SVH** (Netzer et al., 2011) is a digit classification benchmark dataset that contains 600,000 32×32 RGB images of printed digits (from 0 to 9) cropped from pictures of house number plates. **ImageNet** (Deng et al., 2009), also known as ILSVRC 2012, contains 1,000 classes, with 1.28 million training images and 50,000 validation images. For the NLP task, we train on a dataset of 1 million randomly sampled sentences from **Wikipedia articles** (Huggingface, 2021) and evaluate our models on 7 different semantic textual similarity datasets from the SentEval benchmark suite (Conneau & Kiela, 2018): **MR** (movie reviews), **CR** (product reviews), **SUBJ** (subjectivity status), **MPQA** (opinion-polarity), **SST-2** (sentiment analysis), **TREC** (question-type classification), and **MRPC** (paraphrase detection). The **T6SS** effector protein dataset is a public real-world bacteria dataset (SecReT6, (Li et al., 2015)) with actual label scarcity. The sequence length of the genome sample is 1000nt in all experiments.

Tasks We examine and benchmark a model's reliability on different tasks considering evaluation protocols by self-supervised learning (Chen et al., 2020) and Plex's benchmarking tasks (Tran et al., 2022). Specifically, we evaluate our model on the basis of **uncertainty-aware IND generalization**, **OOD detection**, **semi-supervised learning**, **corrupted dataset evaluation** (see Section 5), and **transfer learning to other datasets and tasks** (see Section B.2) .

Evaluation metrics We report prediction/calibration performance with the following metrics, where upward arrows indicate that higher values are desirable, *et vice versa*. **Top-1 accuracy** $\uparrow$: share of test observations for which the correct class is predicted. **AUROC** $\uparrow$: area under the ROC curve arising from different combinations of false-positive and false-negative rates (here: with positive and negative classes referring to being in and out of distribution, respectively) for a gradually increasing classification threshold. **Negative log-likelihood (NLL)** $\downarrow$: negative log-likelihood of test observations under the estimated parameters. **Expected calibration error (ECE)**;(Naeini et al., 2015) $\downarrow$: mean absolute difference between accuracy and confidence (highest posterior probability among predicted classes) across equally-spaced confidence bins, weighted by relative number of samples per bin. **Thresholded adaptive calibration error (TACE)**; (Nixon et al., 2019)) $\downarrow$: modified ECE with bins of equal sample size, rather than equal interval width, and omitting predictions with posterior probabilities falling below a certain threshold (here: 0.01) that often dominate the calibration in tasks with many classes.

Compared methods We compare our method to the following contenders. **Baseline:** self-supervised architectures (i.e., SimCLR, DINO, SCD, or Self-GenomeNet, depending on the task) without probabilistic modification. **Ensemble-SSL**: deep ensemble comprising multiple of the aforementioned baseline networks. **Monte Carlo (MC) dropout**: (Gal & Ghahramani, 2016)

baseline networks with dropout regularization applied during pretraining of baseline encoder. **BatchEnsemble**: baseline encoder with BatchEnsemble applied during pretraining.

5 RESULTS AND DISCUSSION

In-distribution generalization IND generalization (or *prediction calibration*) quantifies how well model confidence aligns with model accuracy. We perform several experiments on small and large image datasets as well as the genomics dataset to evaluate and compare the predictive performance of our proposed model in IND generalization. Here, the base encoder $f_{\boldsymbol{\theta}}$ is frozen after unsupervised pretraining, and the model is trained on a supervised linear classifier. The linear classifier is a fully connected layer followed by softmax, which is placed on top of $f_{\boldsymbol{\theta}}$ after removing the ensemble of sub-networks. High predictive scores and low uncertainty scores are desired.

Figure 3 illustrates the predictive probability of correctness for our model on CIFAR-10, CIFAR-100, ImageNet, and T6SS dataset in terms of Top-1 accuracy, ECE, and NLL, respectively. Based on Figure 3, our method achieves better calibration (ECE and NLL) than the deep ensemble of self-supervised models, MC-dropout, and BatchEnsemble, with substantial margins at large ensemble sizes. More detailed descriptions are provided in Appendix B.1 (Tables 5, 6, 7, and 8).

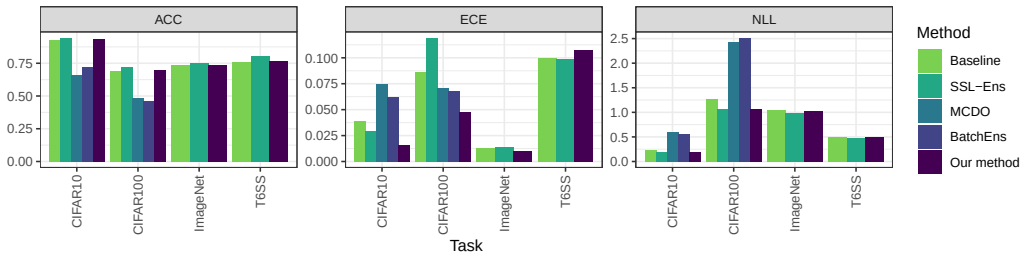


Figure 3: **Uncertainty-awareness: IND generalization** in terms of (a) **top-1 Accuracy** (b) **ECE** (c) **NLL** averaged over in-distribution on test samples of *CIFAR-10/100, ImageNet, T6SS* datasets. Here, we compare our method with BatchEnsemble (BatchEns), MC-dropout (MCDO), ensemble of deep self-supervised networks (SSL-Ens), as well as the baseline. Detailed descriptions of IND generalization for each dataset are presented in Appendix B.1 (Tables 5, 6, 7, and 8).

Out-of-distribution detection OOD detection shows how well a model can recognize test samples from the classes that are unseen during training (Geng et al., 2020). We perform several experiments to compare the model generalization from IND to OOD datasets, and to predict the uncertainty of the models on OOD datasets. Evaluation is performed directly after unsupervised pretraining without a fine-tuning step. Table 2 shows the AUROC on different OOD sets for our model, baseline and deep self-supervised ensemble. Our approach improves over all compared methods.

Table 2: **OOD detection**. Results reported using AUROC show our method enhances the baseline up to 6%.

IND	OOD	Baseline	Ensemble-SSL	Our method
CIFAR-100	SVHN	84.22	84.95	**88.00**
	Uniform	91.65	90.53	**97.57**
	Gaussian	90.00	89.42	**94.10**
	CIFAR-10	74.71	74.80	**75.18**
CIFAR-10	SVHN	95.03	96.68	**97.07**
	Uniform	96.73	91.64	**99.05**
	Gaussian	96.39	93.24	**99.24**
	CIFAR-100	91.79	91.59	**91.87**

Semi-supervised evaluation We explore and compare the performance of our proposed method in the low-data regime. Again, the encoder $f_{\boldsymbol{\theta}}$ is frozen after self-supervised pretraining, and the model is trained on a supervised linear classifier using 1% and 10% of the dataset. The linear classifier is

a fully connected layer followed by softmax. Table 3 shows the result in terms of top-1 accuracy, ECE, and NLL. The results indicate that our method considerably outperforms other methods in the low-data regime – both in terms of calibration and accuracy.

Table 3: **Semi-supervised evaluation**: Top-1 accuracy (ACC), ECE, and NLL for semi-supervised CIFAR-10/100 classification using 1% and 10% training examples.

Method	CIFAR-10 (1%)			CIFAR-10 (10%)			CIFAR-100 (1%)			CIFAR-100 (10%)		
	ACC	ECE	NLL	ACC	ECE	NLL	ACC	ECE	NLL	ACC	ECE	NLL
Baseline	89.1	0.075	0.364	91.1	0.039	0.274	56.2	0.097	2.01	59.5	0.086	1.79
SSL-Ensemble	90.1	0.056	0.334	92.2	0.050	0.257	59.7	0.081	1.86	62.6	0.053	1.48
Our method	90.4	0.018	0.296	92.6	0.016	0.249	59.3	0.060	1.71	62.4	0.042	1.56

Corrupted dataset evaluation Another important component of model robustness is its ability to make accurate predictions when the test data distribution changes. Here, we evaluate model robustness under covariate shift. We employ a configuration similar to the one found in Tran et al. (2022). Figure 4 summarizes the improved performance across metrics of interest. The results confirm that our method outperforms the baseline and achieves comparable predictive performance as a deep self-supervised ensemble – both in terms of calibration (TACE) and AUROC.

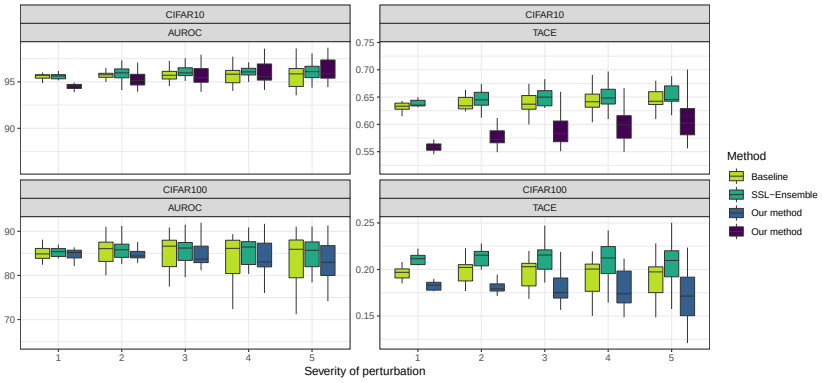


Figure 4: Performance under dataset corruption (CIFAR-10/100 with five levels of increasing perturbation), evaluation in terms of AUROC and TACE for several types of corruption (vertical spread).

6 Ablation Study

In order to build intuition around the behavior and the observed performance of the proposed method, we further investigate the following aspects of our approach in multiple ablation studies exploring: (1) the number M of sub-networks, (2) the role of each component of the proposed loss, and (3) the impact of our approach at the time of pretraining vs. at the finetuning step.

Number of sub-networks We train M individual deep neural networks on top of the representation layer. The networks receive the same inputs but are parameterized with different weights and biases. Here, we provide more details regarding our experiments on IND generalization by considering varying M. Fig. 5a compares the performance in terms of top-1 accuracy, ECE, and NLL for CIFAR-10 and CIFAR-100. Based on the quantitative results depicted in Fig. 5a, the predictive performance improves in both datasets when increasing the number of sub-networks (M) until a certain point. For example, in the case of CIFAR-10, when $M = 3$, our performance is 91.9%; increasing M to 10 levels top-1 accuracy up to 92.6%, while the ECE and NLL decrease from 0.026 and 0.249 to 0.023 and 0.222, respectively. These findings underline that training our sub-networks with a suitable number of heads can lead to a better representation of the data and better calibration.

Analysis of loss The total loss (Eq. 4) is calculated by the combination of self-supervised loss (Eq. 1) and diversity loss (Eq. 3), where the mean value of the embeddings across the ensemble of

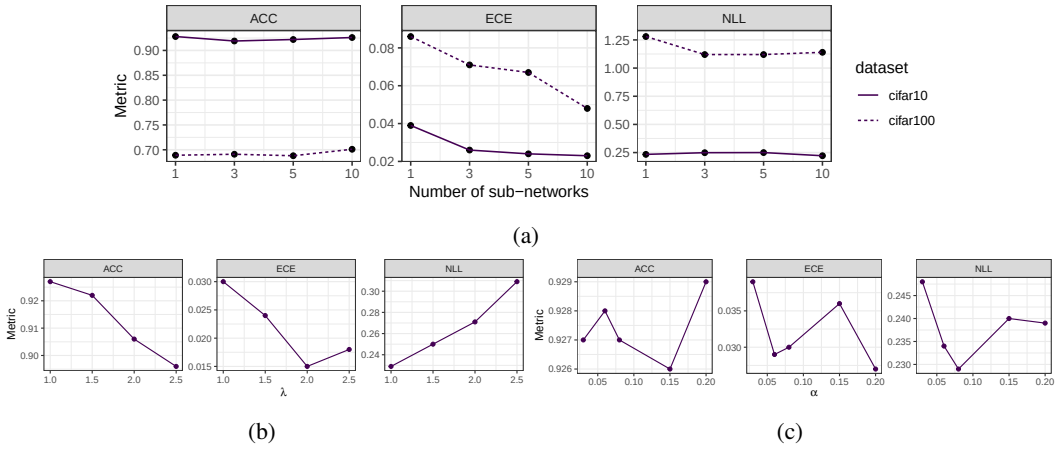


Figure 5: Ablation study on number M of sub-networks (a) and on hyperparameters of our proposed loss (b) λ and (c) α.

sub-networks is fed to the self-supervised loss, and the corresponding standard deviation is used for the diversity loss. First, we note that the use of our diversity regularizer indeed improves uncertainty awareness. The results in Fig. 3 show the impact of our loss function in relation to the baseline. By comparing the first and fifth rows of Table 5, it can be inferred that our proposed loss function results in a much lower ECE (0.016) than the network trained by SimCLR (baseline) with 0.039 on CIFAR-10 dataset. Similarly, the first and third rows of Table 7 compare the predictive probability of correctness of DINO (baseline) and our model on ImageNet.

Second, we explore different hyperparameter configurations to find the optimal values for α and λ in Fig. 5b,5c. Note that, in practice, α and λ must be optimized jointly. The best top-1 accuracy in our case is achieved when α and λ are set to 0.08 and 1.5, respectively, on the CIFAR-10 dataset.

Efficient ensemble of sub-networks at pretraining vs. finetuning We performed additional experiments to study the impact of efficient ensemble and uncertainty-aware loss i) during pretraining, ii) during finetuning, and iii) during both pretraining and finetuning. As shown in Table 4, pretraining with an ensemble of sub-networks is beneficial, and additional fine-tuning with multiple head can further improve performance.

Table 4: **Pretraining vs. Finetuning**: Expected calibration error averaged over uncertainty-aware evaluation on CIFAR-10 datasets.

Method	ACC (%) (↑)	ECE (↓)	NLL (↓)	TACE (↓)
Baseline	92.5	0.039	0.238	0.133
Pretrain-UnSub	92.6	0.032	0.226	0.131
Finetune-UnSub	92.6	0.021	0.222	0.103
Pretrain-UnSub + Finetune-UnSub	92.8	0.023	0.227	0.115

7 Conclusion

In this paper, we presented a novel uncertainty-aware self-supervised framework. We achieved high predictive performance and good calibration using a simple yet effective idea – an ensemble of independent sub-networks that can estimate model uncertainty during pretraining of a self-supervised framework. We introduced a new loss function to encourage diversity among different sub-networks. It is straightforward to add our method to many existing self-supervised learning frameworks during pretraining. Our extensive experimental results show that our proposed method outperforms, or is on par with, an ensemble of self-supervised methods in many different experimental settings.

REFERENCES

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2020.

Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, et al. Big self-supervised models advance medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3478–3488, 2021.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 1298–1312. PMLR, 2022. URL https://proceedings.mlr.press/v162/baevski22a.html.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *ICLR*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Topology preserving local road network estimation from single onboard camera image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17263–17272, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

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

Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/757.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ACL*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

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, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

Nikita Durasov, Timur Bagautdinov, Pierre Baque, and Pascal Fua. Masksembles for uncertainty estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13539–13548, 2021.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.

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

Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631, 2020.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553): 452–459, 2015.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Hüseyin Anil Gündüz, Martin Binder, Xiao-Yin To, René Mreches, Philipp C Münch, Alice C McHardy, Bernd Bischl, and Mina Rezaei. Self-genomenet: Self-supervised learning with reverse-complement context prediction for nucleotide-level genomics data, 2021.

Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schon. Evaluating scalable bayesian deep learning methods for robust computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020.

Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M Dai, and Dustin Tran. Training independent subnetworks for robust prediction. In *ICLR*, 2021.

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*, pp. 770–778, 2016.

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*, pp. 16000–16009, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJz6tiCqYm`.

Yibo Hu and Latifur Khan. Uncertainty-Aware Reliable Text Classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 628–636. ACM, 2021. ISBN 978-1-4503-8332-5. doi: 10.1145/3447548.3467382.

Huggingface. wiki1m_for_simcse.txt, 2021. URL `https://huggingface.co/datasets/princeton-nlp/datasets-for-simcse/blob/main/wiki1m_for_simcse.txt`.

Sergey Ioffe. Batch Renormalization: Towards Reducing Minibatch Dependence in Batch-Normalized Models. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1945–1953, 2017.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A Survey on Contrastive Self-Supervised Learning. *Technologies*, 9(1), 2020.

José Jiménez-Luna, Francesca Grisoni, and Gisbert Schneider. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584, 2020.

Berk Kaya, Suryansh Kumar, Carlos Oliveira, Vittorio Ferrari, and Luc Van Gool. Uncertainty-aware deep multi-view photometric stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12601–12611, 2022.

Tassilo Klein and Moin Nabi. Scd: Self-contrastive decorrelation for sentence embeddings. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Jun Li, Yufeng Yao, H Howard Xu, Limin Hao, Zixin Deng, Kumar Rajakumar, and Hong-Yu Ou. Secret6: a web-based resource for type vi secretion systems found in bacteria. *Environmental microbiology*, 17(7):2196–2202, 2015.

Manling Li, Ruochen Xu, Shuohang Wang, Luowei Zhou, Xudong Lin, Chenguang Zhu, Michael Zeng, Heng Ji, and Shih-Fu Chang. Clip-event: Connecting text and images with event structures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16420–16429, 2022.

Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, and Balaji Lakshminarayanan. Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020),*. arXiv, 2020.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of AAAI'15*, pp. 2901–2907. AAAI Press, 2015.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning, 2011.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring Calibration in Deep Learning, 2019.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Mehmet Ozgur Turkoglu, Alexander Becker, Hüseyin Anil Gündüz, Mina Rezaei, Bernd Bischl, Rodrigo Caye Daudt, Stefano D'Aronco, Jan Dirk Wegner, and Konrad Schindler. Film-ensemble: Probabilistic deep learning via feature-wise linear modulation. *Advances in neural information processing systems*, 35:21271–21284, 2022.

Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. Probabilistic Representations for Video Contrastive Learning. In *CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Vikash Sehwag, Mung Chiang, and Prateek Mittal. SSD: A unified framework for self-supervised outlier detection. *CoRR*, abs/2103.12051, 2021. URL https://arxiv.org/abs/2103.12051.

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. Learning audio-visual speech representation by masked multimodal cluster prediction. *ICLR*, 2022.

Dustin Tran, Jeremiah Liu, Michael W Dusenberry, Du Phan, Mark Collier, Jie Ren, Kehang Han, Zi Wang, Zelda Mariet, Huiyi Hu, et al. Plex: Towards reliability using pretrained large model extensions. *arXiv preprint arXiv:2207.07411*, 2022.

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, Manvel Avetisian, and Leonid Zhukov. Uncertainty Estimation of Transformer Predictions for Misclassification Detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8237–8252. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.566.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020.

Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon Kohl, A. Taylan Cemgil, S. M. Ali Eslami, and Olaf Ronneberger. Contrastive training for improved out-of-distribution detection. *CoRR*, abs/2007.05566, 2020. URL https://arxiv.org/abs/2007.05566.

Cha Zhang and Yunqian Ma. *Ensemble Machine Learning: Methods and Applications*. Springer Publishing Company, Incorporated, 2012. ISBN 1441993258.

Oliver Zhang, Mike Wu, Jasmine Bayrooti, and Noah D. Goodman. Temperature as uncertainty in contrastive learning. *CoRR*, abs/2110.04403, 2021. URL https://arxiv.org/abs/2110.04403.

A IMPLEMENTATION DETAILS

A.1 COMPUTATION COST ANALYSIS

Figure 6 illustrates relative computation cost – as compared to the baseline – in terms of the number of parameters, computation time, and memory required between our model and a deep self-supervised ensemble.

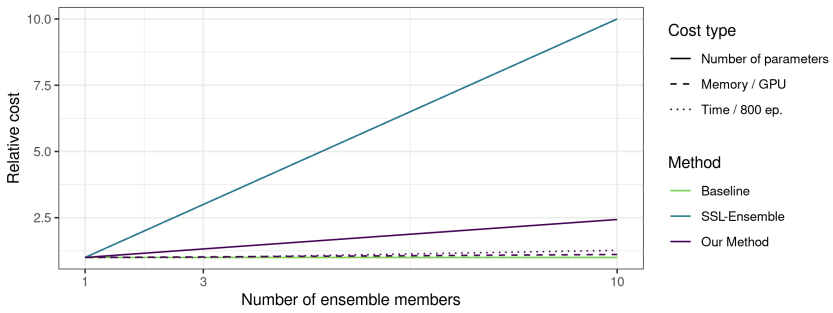


Figure 6: The test time cost (purple, dotted) and memory cost (purple, dashed) of our model w.r.t. the ensemble size. The figures are relative to the cost incurred by a single model (green). Inference time cost and memory cost of a deep self-supervised ensemble are plotted in blue.

A.2 DATA AUGMENTATION FOR COMPUTER VISION DATASETS

We define a random transformation function $\boldsymbol{T}$ that applies a combination of crop, horizontal flip, color jitter, and grayscale. Similar to Chen et al. (2020), we perform crops with a random size from 0.2 to 1.0 of the original area and a random aspect ratio from $3/4$ to $4/3$ of the original aspect ratio. We also apply horizontal mirroring with a probability of 0.5. Then, we apply grayscale with a probability of 0.2 as well as color jittering with a probability of 0.8 and a configuration of $(0.4, 0.4, 0.4, 0.1)$. However, for ImageNet, we define augmentation based on the original DINO from their official repository. In all experiments, at the testing phase, we apply only resize and center crop.

A.3 HYPERPARAMETERS FOR SELF-SUPERVISED NETWORK ARCHITECTURES

SimCLR (Chen et al., 2020): we use ResNet-50 as a backbone, a loss temperature of 0.07, batch size 512, and a cosine-annealing learning rate scheduler. The embedding size is 2048, and we train for 800 epochs during pretraining. **DINO** (Caron et al., 2021): we use ViT-small as a backbone, patch size 16, batch size 1024, and a cosine-annealing learning rate scheduler. The embedding size is 384/1536, and we train for 100 epochs during pretraining.

B ADDITIONAL RESULTS

B.1 UNCERTAINTY-AWARENESS IND GENERALIZATION

Tables 5, 6, and 7 present results for the predictive performance and calibration of our model on CIFAR-10, CIFAR-100, and ImageNet, respectively. Based on Table 5, our method achieves better calibration than the deep ensemble of self-supervised networks, MC-dropout, and BatchEnsemble, with substantial margins at large ensemble sizes.

Table 5: **Uncertainty-Awareness: IND Generalization**: Top-1 accuracy, ECE and NLL averaged over in-distribution on test samples of the **CIFAR-10** dataset. The best score for each metric is shown in **bold**, and the second-best is underlined.

Method	Top-1 Acc (%) (↑)			ECE (↓)			NLL (↓)		
# member (M)	3	5	10	3	5	10	3	5	10
Baseline		92.8 ± 0.4			0.039 ± 0.002			0.233 ± 0.011	
SSL-Ensemble	92.8 ± 0.1	93.0 ± 0.2	$\mathbf{94.2 \pm 0.3}$	0.043 ± 0.02	0.033 ± 0.01	$\underline{0.029 \pm 0.02}$	0.221 ± 0.011	0.226 ± 0.009	$\underline{0.199 \pm 0.004}$
MC Dropout	65.7 ± 0.2	66.3 ± 0.2	66.4 ± 0.2	0.083 ± 0.014	0.077 ± 0.009	0.075 ± 0.005	0.66 ± 0.012	0.637 ± 0.002	0.593 ± 0.006
BatchEnsemble	69.1 ± 04	72.1 ± 0.3	71.9 ± 0.2	0.064 ± 0.011	0.061 ± 0.008	0.062 ± 0.005	$0.613 \pm xx$	0.58 ± 0.007	0.551 ± 0.004
Our method	92.6 ± 0.2	92.9 ± 0.1	$\underline{93.6 \pm 0.1}$	0.021 ± 0.004	0.019 ± 0.002	$\mathbf{0.016 \pm 0.001}$	0.241 ± 0.010	0.221 ± 0.005	$\mathbf{0.193 \pm 0.003}$

Table 6: **Uncertainty-Awareness: IND Generalization**: Top-1 accuracy, ECE and NLL averaged over in-distribution on test samples of the **CIFAR-100** dataset. The best score for each metric is shown in **bold**, and the second-best is underlined.

Method	Top-1 Acc (%) (↑)			ECE (↓)			NLL (↓)		
# member (M)	3	5	10	3	5	10	3	5	10
Baseline		68.9 ± 0.3			0.086 ± 0.014			1.28 ± 0.05	
SSL-Ensemble	70.6 ± 0.12	71.4 ± 0.5	$\mathbf{72.0 \pm 0.2}$	0.12 ± 0.01	0.122 ± 0.01	0.119 ± 0.04	1.09 ± 0.01	1.12 ± 0.01	$\underline{1.06 \pm 0.02}$
MC Dropout	46.3 ± 0.1	45.2 ± 0.4	48.2 ± 0.1	0.077 ± 0.012	0.081 ± 0.002	0.071 ± 0.002	2.66 ± 0.11	2.37 ± 0.02	2.43 ± 0.06
BatchEnsemble	44.1 ± 03	45.2 ± 0.2	46.1 ± 0.1	0.073 ± 0.01	0.071 ± 0.08	$\underline{0.068 \pm 0.001}$	2.43 ± 0.03	2.64 ± 0.007	2.51 ± 0.004
Our method	67.7 ± 0.1	68.8 ± 0.1	$\underline{70.1 \pm 0.0}$	0.067 ± 0.001	0.063 ± 0.001	$\mathbf{0.048 \pm 0.000}$	0.114 ± 0.005	0.116 ± 0.0002	$\mathbf{1.06 \pm 0.001}$

Table 7: **Uncertainty-Awareness: IND Generalization**: Top-1 accuracy, ECE and NLL averaged over in-distribution on test samples of the **ImageNet** dataset. The best score for each metric is shown in **bold**, and the second-best is underlined.

Method	Top-1 Acc (%) (↑)	ECE (↓)	NLL (↓)
Baseline	73.8 ± 0.3	0.013 ± 0.015	1.05 ± 0.01
SSL-Ensemble	$\mathbf{75.1 \pm 0.1}$	$\underline{0.014 \pm 0.000}$	$\mathbf{0.98 \pm 0.01}$
Our method	$\underline{74.0 \pm 0.0}$	$\mathbf{0.010 \pm 0.000}$	$\underline{1.03 \pm 0.01}$

Table 8: **Uncertainty-Awareness: IND Generalization**: Top-1 accuracy, ECE and NLL averaged over in-distribution on test samples of the **T6SS Identification** dataset. The best score for each metric is shown in **bold**, and the second-best is underlined.

Method	Top-1 Acc (%) (↑)	ECE (↓)	NLL (↓)
Baseline	75.9 ± 2.0	$\underline{0.100 \pm 0.006}$	0.502 ± 0.020
SSL-Ensemble	$\mathbf{80.2 \pm 0.7}$	$\mathbf{0.099 \pm 0.014}$	$\mathbf{0.471 \pm 0.011}$
Our method	$\underline{76.7 \pm 2.3}$	0.108 ± 0.006	$\underline{0.492 \pm 0.024}$

We also performed experiments on a dataset of 1-dimensional genomic sequences – the T6SS identification of effector proteins– to demonstrate that uncertainty-aware subnetworks can also be readily combined with existing models for 1-dimensional datasets and models. Based on Table 8, our method improves the accuracy and the calibration compared to the baseline.

B.2 TRANSFER TO OTHER TASKS AND DATASET

We further assess the generalization capacity of the learned representation on learning a new task in NLP. We train our model without any labels on a dataset of sentences from Wikipedia (Huggingface, 2021) and fine-tune the pretrained representation on seven different semantic textual similarity datasets from the SentEval benchmark suite (Conneau & Kiela, 2018): **MR** (movie reviews), **CR** (product reviews), **SUBJ** (subjectivity status), **MPQA** (opinion-polarity), **SST-2** (sentiment analysis), **TREC** (question-type classification), and **MRPC** (paraphrase detection). Then, we evaluate on the test set of each dataset. Figure 7 provides a comparison of transfer learning performance of our self-supervised approach for different tasks. Our results in Figure 7 indicate that our approach performs comparably to or better than the baseline method.

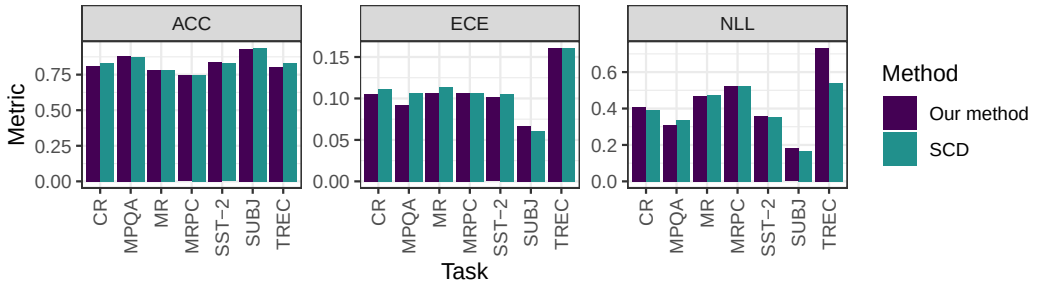


Figure 7: **Transfer to other dataset and tasks:** Comparision of Sentence embedding performance on semantic textual similarity tasks .

We test the performance of the trained model on ImageNet (Deng et al., 2009) on CIFAR-10 Krizhevsky (2009) dataset where the model is trained for 100 epochs.

Table 9: **Transfer to other dataset**: Expected calibration error averaged over uncertainty-aware evaluation on CIFAR-10 datasets.

Method	ACC (%) (↑)	ECE (↓)	NLL (↓)	TACE (↓)
Baseline	73.5	0.038	0.78	0.20
Our method	73.9	0.030	0.75	0.18