Great Models Think Alike: Improving Model Reliability via Inter-Model Latent Agreement

Ailin Deng [1] **Miao Xiong** [2] **Bryan Hooi** [1 2]

Abstract

Reliable application of machine learning is of primary importance to the practical deployment of deep learning methods. A fundamental challenge is that models are often unreliable due to overconfidence (Hendrycks & Gimpel, 2017). In this paper, we estimate a model's reliability by measuring *the agreement between its latent space, and the latent space of a foundation model*. However, it is challenging to measure the agreement between two different latent spaces due to their incoherence, e.g., arbitrary rotations and different dimensionality. To overcome this incoherence issue, we design a *neighborhood agreement measure* between latent spaces and find that this agreement is surprisingly well-correlated with the reliability of a model's predictions. Further, we show that fusing neighborhood agreement into a model's predictive confidence in a post-hoc way significantly improves its reliability. Theoretical analysis and extensive experiments on failure detection across various datasets verify the effectiveness of our method on both in-distribution and out-of-distribution settings.

1. Introduction

Model reliability is a critical and challenging issue in deep neural networks for deploying neural network systems in real-world applications, particularly in safety-critical domains such as autonomous driving and medical diagnosis. In particular, a key challenge is that models tend to be overconfident (Guo et al., 2017; Hendrycks & Gimpel, 2017; Ovadia et al., 2019), and it is hard to identify such overconfidence purely based on the model's own internal states, as these internal states are themselves potentially unreliable.

A possible answer to this dilemma comes from the recent emergence of powerful general purpose (or 'foundation') models (Brown et al., 2020; Bommasani et al., 2021; Radford et al., 2021), which provide rich "implicit knowledge" while being freely available for use without additional training costs. This presents an opportunity to use them to assist in evaluating the reliability of a newly trained model.

As encapsulated by the phrase "great minds think alike", it is intuitive that a model tends to be more reliable on some input if its reasoning aligns well with that of other models. For example, if we want to examine whether a human learner has well-understood some input image (e.g. an animal), we can ask them questions about it (e.g. what animals is it similar to?): the more they agree with other human learners, the more confident we can be that they have correctly understood the image. Similarly, for models, we want to evaluate the reliability of a model by estimating the extent to which its reasoning agrees with that of a foundation model. This further leads to the main challenge: how can we quantitatively measure how much two models agree on a concept? In this work, we propose to measure this model agreement via latent spaces. Intuitively, the two models agree if their latent spaces "model the concept similarly"; however, this is complicated by the fact that different models are typically trained with different data distributions, model architectures and optimization objectives, leading to incoherence between different latent spaces: e.g. differing by an arbitrary rotation, and different dimensionalities.

To solve this problem, we introduce *inter-model latent agreement* - a framework for measuring of how much two models agree on a sample while avoiding the incoherence issue, and then show its utility for estimating and enhancing model reliability. Our framework uses the similarity of neighborhoods in the latent spaces of an input as a proxy task to measure agreement between latent spaces of different models.

In particular, we present our main empirical observation that the reliability (or probability of correctness) of a model on a sample correlates well with its inter-model latent agreement with foundation models on that sample. Motivated by this, we propose our latent space agreement framework, which exploits the inter-model agreement to improve prediction reliability in a post-hoc way. Notably, our proposed

[1]School of Computing, National University of Singapore, Singapore [2]Institute of Data Science, National University of Singapore, Singapore. Correspondence to: Ailin Deng <ailin@u.nus.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

inter-model latent agreement can be measured with only unlabeled samples, making it more broadly applicable. We further verify the effectiveness of our framework by conducting extensive experiments on failure detection over various datasets and provide theoretical analysis for our method.

Overall, the contributions and benefits of our approach are as follows [1]:

- (Empirical Findings) We show that the inter-model agreement highly correlates with classification accuracy, suggesting the value of latent space agreement to improve a newly trained model's reliability.
- (Generality) We propose a general framework that enables using any foundation model via latent spaces in a post-hoc way without any fine-tuning to improve the predictive reliability of a newly trained model.
- (Effectiveness) We quantitatively verify the performance of our framework on failure detection across various datasets, including large-scale in-distribution (ID) and out-of-distribution (OOD) datasets, and provide further exploration, empirical and theoretical justification of the framework.

2. Related Work

2.1. Failure Detection

The main goal of failure detection is to predict whether a trained classifier will make an error on a test sample (Jaeger et al., 2023). Hendrycks & Gimpel 2017 propose maximum softmax probabilty (MSP), which directly uses the softmax predictions of the trained model. Follow-up works propose other uncertainty measures from a trained model, based on Monte Carlo Dropout or aggregated from multiple trained models, e.g., predictive entropy or variants (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017; Liu et al., 2020). Jiang et al. 2018 propose a distance ratio in the latent space of a classifier. In addition to detecting failures in in-distribution data, detecting failures under distribution shifts is a crucial problem to enhance model reliability in real-world applications, and has received increasing attention (Hendrycks & Dietterich, 2019; Koh et al., 2021; Vaze et al., 2022). Previous works have aimed to utilize internal information from a trained classifier (Xiong et al., 2022; Deng et al., 2022). However, a classifier itself can be potentially unreliable (Guo et al., 2017; Hein et al., 2019), which motivates us to use external information to validate the reliability of a prediction and further enhance failure detection.

[1]Our code is available via https://github.com/d-ailin/latent-agreement

2.2. Foundation Models

The recent powerful foundation models (Radford et al., 2021; Bommasani et al., 2021) are pretrained on large-scale data and can provide rich "implicit knowledge" to validate the prediction from a newly trained classifier. The prevalent paradigm to use these foundation models is fine-tuning with downstream data. However, traditional fine-tuning is often computationally intensive and requires a large amount of downstream labeled data, particularly as foundation models continue to grow in size. To alleviate these issues, recent methods propose to adapt the foundation models before use for some applications by prompting, instead of fully fine-tuning (Bommasani et al., 2021). Our method is different from the previous works as we propose to use the agreement between the latent space of a trained classifier and the latent space of a foundation model to validate a prediction, which requires no fine-tuning or adaptation.

3. Proposed Method

3.1. Preliminaries

Let $D = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{n}$, denote the training dataset containing n samples, where $\mathbf{x}^{(i)} \in \mathbb{R}^m$ is the i-th input sample and $y^{(i)} \in \mathcal{Y} = \{1, \dots, C\}$ is the corresponding true class. A classification model consists of two parts: a feature extractor $B : \mathcal{X} \to \mathbb{R}^d$ and a linear head $f_w : \mathbb{R}^d \to \mathbb{R}^C$, parameterized by w. Given an input $\mathbf{x}$, the model produces a latent feature vector $\mathbf{z} = B(\mathbf{x})$ followed by the softmax probability output and predictive label:

$$\hat{P}(Y \mid \mathbf{x}, B, w) = \mathsf{softmax}(f_w(\mathbf{z})) \tag{1}$$

$$\hat{y} = \underset{c \in \mathcal{Y}}{\operatorname{argmax}}\, \hat{P}(Y = c \mid \mathbf{x}, B, w). \tag{2}$$

Given an input $\mathbf{x}$, a foundation model can also produce a latent feature vector $\mathbf{h} = H(\mathbf{x}) \in \mathbb{R}^h$ where $H : \mathcal{X} \to \mathbb{R}^h$. For example, H can be an image encoder from a multi-modal foundation model (Radford et al., 2021).

3.2. Problem Definition

Failure Detection Also known as misclassification or error prediction (Hendrycks & Gimpel, 2017), failure detection aims to predict if a trained model makes an erroneous prediction on a test sample. In general, it requires a score for any given sample's prediction, where a lower score implies that the prediction is more likely to be wrong.

For a standard network, the baseline method is to use maximum softmax output as the confidence score for failure detection (Hendrycks & Gimpel, 2017; Ovadia et al., 2019):

$$\hat{p} \coloneqq \hat{P}(Y = \hat{y} \mid \mathbf{x}, B, w) \tag{3}$$

However, merely relying on the obtained confidence score from a newly trained classifier can be unsafe due to the

overconfidence issue (Hendrycks & Gimpel, 2017; Guo et al., 2017) and this concern is even more pronounced under distribution shifts (Ovadia et al., 2019; Hendrycks & Dietterich, 2019; Taori et al., 2020). We thus propose to employ information from foundation models to improve model reliability, instead of only using the information from the trained model.

3.3. Inter-Model Latent Agreement Framework

Overview We propose an inter-model latent agreement framework to compute the agreement scores based on latent spaces and use it as an auxiliary source of information to boost the failure detection performance. The framework involves two steps: 1) measuring agreement between a newly trained model and a foundation model on samples; 2) fusing the agreement information into the predictive confidence in a post-hoc way via input-dependent temperature scaling.

Measuring Inter-Model Latent Agreement Neural network models usually first project input data into latent space, then perform classification or generation based on the latent spaces. Thus, latent spaces are informative and can be taken as the network's capacity for capturing the information in input samples. In this work, we aim to measure latent space agreement between models to represent their agreement.

Given an encoder B from a trained classifier and a pretrained encoder H from a foundation model, to estimate how reliable the trained classifier is on a sample $\mathbf{x}$, we aim to estimate the agreement between the models' latent spaces around $\mathbf{x}$. Specifically, we compute feature vectors $\mathbf{z} = B(\mathbf{x})$ and $\mathbf{h} = H(\mathbf{x})$ with the encoder B and pretrained encoder H, respectively. However, as these latent spaces can be incoherent, e.g., differing by an unknown rotation, or different dimensions of latent spaces ($d \neq h$), this makes explicit distance comparison between $\mathbf{z} \in \mathbb{R}^d$ and $\mathbf{h} \in \mathbb{R}^h$ unsuitable.

To overcome this incompatibility, we compare *neighborhoods* (e.g. nearest neighbors and distances to them) between two latent spaces around a sample as a surrogate task to measure the agreement instead of a direct distance measure between $\mathbf{z}$ and $\mathbf{h}$. For example, if two latent spaces are identical after a rotation transformation, the neighborhoods of a sample in the two latent spaces must be the same. Conversely, if the neighborhoods of a sample in the two latent spaces are similar, the two latent spaces around this sample are expected to be similar through some unknown transformation, due to the high level of agreement between these two latent spaces.

Specifically, denote the test sample as $\mathbf{x}^{\text{test}}$, the encoder B from a classifier and the classifier's training dataset $D = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^n$. We obtain the feature vectors $\mathbf{z}^{\text{test}} := B(\mathbf{x}^{\text{test}})$ and $\mathbf{z}_i := B(\mathbf{x}^{(i)})$ for $1 \leq i \leq n$. We denote:

$$\mathbb{Z} := (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n). \tag{4}$$

Similarly, we repeat this process with the pretrained encoder H to get the feature vectors $\mathbf{h}^{\text{test}} := H(\mathbf{x}^{\text{test}})$ and $\mathbf{h}_i := H(\mathbf{x}^{(i)})$ for $1 \leq i \leq n$. We denote:

$$\mathbb{H} := (\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n). \tag{5}$$

To represent the ranking of training samples based on their similarity to the test sample in the latent space, we use a permutation generation function G to produce a permutation containing the indexes of feature vectors in the training feature set $\mathbb{Z}$, ordered from nearest to farthest distance from the test feature vector $\mathbf{z}^{\text{test}}$:

$$\begin{aligned} G(\mathbf{z}^{\text{test}}, \mathbb{Z}) :=& (\Pi_{(1)}, \Pi_{(2)}, \ldots, \Pi_{(n)}) \\ \text{s.t.} \quad & s(\mathbf{z}^{\text{test}}, \mathbf{z}_{\Pi_{(1)}}) \geq s(\mathbf{z}^{\text{test}}, \mathbf{z}_{\Pi_{(2)}}) \geq s(\mathbf{z}^{\text{test}}, \mathbf{z}_{\Pi_{(n)}}), \end{aligned} \tag{6}$$

where we use the cosine similarity function as s. Similarly, we get another permutation using $\mathbf{h}^{\text{test}}$ and $\mathbb{H}$ for the same test sample $\mathbf{x}^{\text{test}}$ based on the pretrained encoder H. We use $\Pi^* := G(\mathbf{z}^{\text{test}}, \mathbb{Z})$ and $\Pi' := G(\mathbf{h}^{\text{test}}, \mathbb{H})$ to represent the permutations obtained from the encoder B from the classifier and the pretrained encoder H, respectively.

Next, to measure the similarity between two permutations Π^* and Π', we introduce Normalized Discounted Cumulative Gain (NDCG), a ranking quality measure.

Definition 3.1. (Normalized Discounted Cumulative Gain (NDCG) (Järvelin & Kekäläinen, 2002)). Given a ranking Π^* and another ranking Π', let r denote our importance scoring function, where $r(i)$ outputs the importance score of the i-th sample, and $r(\Pi^*_{(1)}) \geq r(\Pi^*_{(2)}) \geq \cdots \geq r(\Pi^*_{(n)})$:

$$\mathsf{NDCG}(\Pi^*, \Pi', r) := \frac{\sum_i^n \frac{r(\Pi'_{(i)})}{\log(i+1)}}{\sum_i^n \frac{r(\Pi^*_{(i)})}{\log(i+1)}}. \tag{7}$$

The NDCG values range from 0 to 1 after normalization. Intuitively, the NDCG metrics quantitatively evaluate the ranking quality of Π' compared to the ranking Π^*, considering the importance scoring function r and ranking position penalty with logarithmic discounting function. Note that any importance scoring function r which satisfies the requirement of producing decreasing values according to the perfect ranking is plausible. In particular, r can be a function outputting 0 and 1 depending on whether the training sample is one of the k-nearest training samples or not:

$$r(i) := \mathbb{1}(\mathbf{z}_i \in \mathcal{N}_{\mathbb{Z},k}(\mathbf{z}^{\text{test}})). \tag{8}$$

It means that we treat the nearest k samples as most important, and we can control the neighborhood size with k.

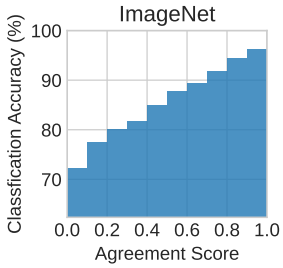


Figure 1. Positive correlation between agreement score and classification accuracy. The agreement score is calculated based on the ImageNet classifier (ViT-B/16) and CLIP ViT/L-14.

Wang et al. 2013 shows that this choice of importance scoring function also provides certain consistency guarantees.

As such, given the encoder B from the trained classifier and pretrained encoders $H_1, \ldots, H_m$ from m different foundation models, we formally define our inter-model latent agreement score as follows:

Definition 3.2. (Inter-model Latent Agreement Score). Given a test sample $\mathbf{x}^{\text{test}}$, the training samples $\{\mathbf{x}^{(i)}\}_{i=1}^{n}$, the encoder B from a trained classifier and pretrained encoders $\mathcal{H} := \{H_1, \ldots, H_m\}$ from m foundation models, recall $\mathbb{Z} := (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)$ and let $\mathbb{H}^i = (H_i(\mathbf{x}^{(1)}), H_i(\mathbf{x}^{(2)}), \ldots, H_i(\mathbf{x}^{(n)}))$, the inter-model latent agreement score is:

$$\mathsf{AS}(\mathbf{x}^{\text{test}}, B, \mathcal{H}) := \frac{1}{m}\sum_{i}^{m} \mathsf{NDCG}(\Pi^*, \Pi^i, r), \quad (9)$$

where $\Pi^* := G(\mathbf{z}^{\text{test}}, \mathbb{Z})$ and $\Pi^i := G(H_i(\mathbf{x}^{\text{test}}), \mathbb{H}^i)$.

The definition indicates that given a test sample, we average the ranking agreement across different foundation models as the inter-model latent agreement score.

3.4. Main Empirical Observation

From Figure 1, we observe that the agreement score has a clear positive correlation with the classification accuracy. This empirical evidence indicates that a prediction which has a higher latent space agreement with a foundation model tends to be predicted correctly in the classification task. This validates our use of agreement scores for assessing the reliability of a prediction and further improving the original predictive confidence to detect failure.

3.5. Input-based Temperature Scaling

As we aim to adjust the predictive confidence to improve failure detection performance but without altering the classifier's final predicted label, an appealling way is input-dependent temperature scaling, which is an extension of temperature scaling (Guo et al., 2017; Deng et al., 2022). Classical temperature scaling uses a single scalar temperature parameter t to rescale the softmax distribution. Using our agreement score for each sample $\mathbf{x}$ as prior information, we propose to obtain a scalar temperature $\tau(\mathbf{x})$ as a learned function of the agreement score $\mathsf{AS}(\mathbf{x}, B, \mathcal{H})$ based on Definition 3.2:

$$\tau(\mathbf{x}) := t + t_s \mathsf{AS}(\mathbf{x}, B, \mathcal{H}) \quad (10)$$

$$\tilde{P}(Y \mid \mathbf{x}) := \mathsf{softmax}\left(\frac{f_w(\mathbf{z})}{\tau(\mathbf{x})}\right) \quad (11)$$

Here, $\tilde{P}(Y \mid \mathbf{x})$ contains our output calibrated probabilities. t and t_s are learnable parameters; they are optimized via negative likelihood loss on the validation set, similarly to in classical temperature scaling (Guo et al., 2017). For each sample $\mathbf{x}$, we obtain $\tau(\mathbf{x})$ as its input-dependent temperature. With $\tau(\mathbf{x}) = 1$, we recover the original predicted probabilities $\hat{p}$ for the sample. As all logit outputs of a sample are divided by the same scalar, the predictive label is unchanged. In this way, we calibrate the softmax distribution based on the agreement score, without compromising the model's accuracy. Note that though temperature scaling was mainly proposed for calibration, recent findings show temperature scaling can also improve failure detection (Galil et al., 2023) by recalibrating the predictive probability distribution with proper scoring rules, which encourages both calibration and ranking for predictive confidence (Gneiting et al., 2007; Kuleshov & Deshpande, 2022).

We summarize our framework in Algorithm 1 in Appendix.

4. Experiments

In this section, we conduct experiments to answer the following research questions:

- (Performance in ID: Section 4.2) How well does our method perform on in-distribution failure prediction compared to the baseline methods?
- (Exploration Study: Section 4.2) How do different foundation models affect our failure detection performance? How does this relate to the model family used?
- (Performance in OOD: Section 4.3) How does it perform under OOD, i.e. distribution shifts?
- (Case Study: Section 4.4) Can our method provide plausible explanation/visualization for samples with high/low agreement scores?
- (Ablation Study: Section 4.5) How sensitive is the method to different hyperparameters? How does it perform when using other similarity measures?

4.1. Experimental Setup

Baselines Our baseline methods include the Maximum Softmax Probability (MSP) (Hendrycks & Gimpel, 2017),

other uncertainty measure: Entropy and Energy (Lakshminarayanan et al., 2017; Liu et al., 2020), a distance based measure: TrustScore (Jiang et al., 2018), MaxLogit proposed for distribution shift (Vaze et al., 2022) and vanilla Temperature Scaling (T.S.) (Guo et al., 2017).

For details on datasets, classifiers, foundation/pretrained models, and experimental protocols, see Appendix A.

4.2. Failure Detection in ID data

We first evaluate failure detection performance in in-distribution (ID) data and further explore the effect of using different pretrained models. We also study the correlation between each pretrained model's performance and its corresponding KNN accuracy in the same dataset. We also find strong correlation in performance within each model family.

Performance Evaluation The reported single-model result uses the model with the best ImageNet accuracy in our model candidate pool, CLIP ViT/L-14. For multiple-model settings, we adopt the models with top 2 ImageNet accuracy: CLIP ViT/L-14 and ViT/L-16 (ImageNet-21K). We conduct further analysis about the effect of different pretrained models later.

As shown in Table 1, our method can outperform the baseline methods over different datasets, including large-scale dataset, ImageNet, for both CNN and ViT classifiers. Our empirical result confirms the previous findings that ViT can generally perform better than CNN classifiers in accuracy on uncertainty estimation-related tasks (Fort et al., 2021; Minderer et al., 2021; Galil et al., 2022).

Effect of Different Foundation Models Given the diversity of foundation models, which can vary in terms of architecture, training data, and optimization losses, it is important to further investigate the impact of different pretrained models on our proposed method.

We demonstrate the average performance over base models and datasets (excluding ImageNet[2]) for different pretrained models in Figure 2, which shows that every pretrained model in our model candidate pool can surpass MSP on average. We compare the performance of models pretrained on datasets of increasing size: ImageNet-1K, ImageNet-21K, and CLIP WebImageText dataset (400 million image-text pairs). Echoing the previous findings in transfer learning (Kolesnikov et al., 2020), we also observe a performance boost of models of larger size pretrained on larger datasets. With the similar model size and pretrained on the same dataset, ViTs generally perform better than CNNs, except for the cases where the dataset is in a smaller

[2]As some pretrained models are trained with ImageNet samples, we exclude ImageNet dataset.

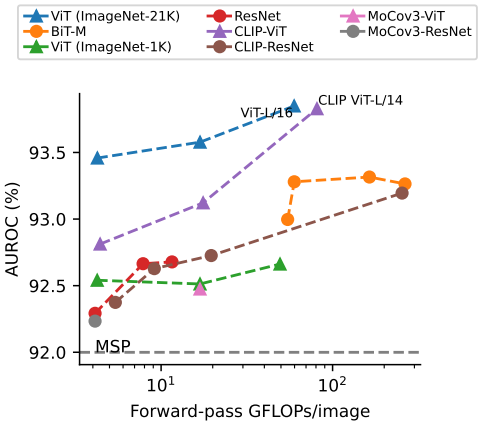


Figure 2. Performance of single-model with each pretrained model average over in-distribution datasets. x-axis: Inference GPU cost. See Appendix C.2 for plots for different base models.

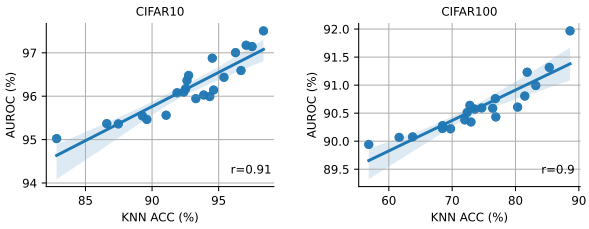


Figure 3. Strong correlation between failure detection performance and KNN accuracy for each pretrained model. See Figure 9 for plots for other datasets.

scale, e.g., ImageNet-1K. The self-supervised pretrained models, e.g. MoCov3 ViT/ResNet, achieve comparable results with models pretrained with supervision signals. Similar to the previous findings in (Kornblith et al., 2019b), which suggests that better performing models transfer better to the downstream tasks, we find the failure detection performance of pretrained models related to these models' accuracy performance in ImagetNet-1K, as shown in Figure 8. Thus, we suggest to select the pretrained model with the highest accuracy on ImageNet-1K for failure detection, without any prior knowledge about the trained models, e.g. the training datasets.

Correlation Between Performance and KNN Accuracy We further investigate the failure detection performance for each pretrained model in a particular dataset. Note that we use KNN classifier (Wu et al., 2018; Caron et al., 2021), a simple weighted nearest neighbor classifier, as a performance proxy to evaluate the pretrained model's performance on the downstream task (Renggli et al., 2022). Figure 3 and 9 show the strong correlation between the failure detection performance and the KNN classifier performance

Table 1. AUROC (%) averaged over 6 runs. The base models are finetuned based on pretrained models: ResNet-50 or ViT-B/16. ViT for ImagNet is fine-tuned on CLIP VIT-B/16 model. AS: inter-model latent agreement score in Eq(9). single: uses CLIP ViT/L-14 as foundation model. multiple: uses CLIP ViT/L-14 and ViT/L-16 (ImageNet-21K) as foundation models. The best result is bolded.

Model	Method	CIFAR10	CIFAR100	STL	BIRDS	FOOD	ImageNet
CNN	MSP	94.15$_{\pm 0.06}$	88.42$_{\pm 0.21}$	95.73$_{\pm 0.35}$	85.97$_{\pm 0.76}$	88.79$_{\pm 0.19}$	86.19$_{\pm 0.07}$
	Entropy	94.14$_{\pm 0.07}$	88.45$_{\pm 0.22}$	95.62$_{\pm 0.36}$	84.83$_{\pm 0.88}$	88.78$_{\pm 0.21}$	84.05$_{\pm 0.06}$
	Energy	90.83$_{\pm 0.32}$	83.84$_{\pm 0.29}$	94.31$_{\pm 0.64}$	77.72$_{\pm 2.10}$	83.83$_{\pm 0.19}$	72.72$_{\pm 0.11}$
	MaxLogit	91.00$_{\pm 0.32}$	84.20$_{\pm 0.30}$	94.44$_{\pm 0.63}$	79.82$_{\pm 1.76}$	84.59$_{\pm 0.20}$	76.49$_{\pm 0.11}$
	TrustScore	95.84$_{\pm 0.24}$	89.13$_{\pm 0.20}$	97.70$_{\pm 0.20}$	84.45$_{\pm 1.27}$	85.71$_{\pm 0.26}$	75.44$_{\pm 1.19}$
	T.S.	93.76$_{\pm 0.14}$	87.21$_{\pm 0.22}$	95.54$_{\pm 0.37}$	85.88$_{\pm 0.79}$	88.58$_{\pm 0.19}$	86.35$_{\pm 0.07}$
	T.S. (w/ AS, single)	97.45$_{\pm 0.05}$	89.63$_{\pm 0.22}$	99.32$_{\pm 0.12}$	88.27$_{\pm 0.71}$	**92.33**$_{\pm 0.14}$	86.38$_{\pm 0.08}$
	T.S. (w/ AS, multiple)	**98.07**$_{\pm 0.08}$	**91.04**$_{\pm 0.24}$	**99.33**$_{\pm 0.13}$	**89.10**$_{\pm 0.65}$	92.17$_{\pm 0.16}$	**86.39**$_{\pm 0.09}$
ViT	MSP	96.39$_{\pm 0.44}$	92.66$_{\pm 0.16}$	98.64$_{\pm 0.40}$	88.23$_{\pm 0.45}$	92.77$_{\pm 0.26}$	85.42$_{\pm 0.09}$
	Entropy	96.36$_{\pm 0.44}$	92.52$_{\pm 0.17}$	98.60$_{\pm 0.40}$	87.58$_{\pm 0.45}$	92.83$_{\pm 0.27}$	81.92$_{\pm 0.09}$
	Energy	91.56$_{\pm 0.76}$	83.67$_{\pm 0.52}$	93.52$_{\pm 1.24}$	76.73$_{\pm 0.72}$	87.00$_{\pm 0.32}$	65.81$_{\pm 0.11}$
	MaxLogit	91.71$_{\pm 0.73}$	84.49$_{\pm 0.45}$	94.15$_{\pm 1.21}$	79.31$_{\pm 0.72}$	87.67$_{\pm 0.33}$	74.48$_{\pm 0.13}$
	TrustScore	97.14$_{\pm 0.31}$	92.33$_{\pm 0.13}$	99.32$_{\pm 0.20}$	87.49$_{\pm 0.61}$	89.03$_{\pm 0.46}$	82.49$_{\pm 0.58}$
	T.S.	96.11$_{\pm 0.50}$	92.31$_{\pm 0.12}$	98.63$_{\pm 0.40}$	88.11$_{\pm 0.46}$	92.83$_{\pm 0.25}$	86.44$_{\pm 0.09}$
	T.S. (w/ AS, single)	96.84$_{\pm 0.31}$	92.83$_{\pm 0.17}$	**99.46**$_{\pm 0.18}$	88.78$_{\pm 0.47}$	**93.81**$_{\pm 0.24}$	86.97$_{\pm 0.10}$
	T.S. (w/ AS, multiple)	**97.36**$_{\pm 0.25}$	**93.14**$_{\pm 0.15}$	99.34$_{\pm 0.28}$	**88.98**$_{\pm 0.47}$	93.64$_{\pm 0.24}$	**87.36**$_{\pm 0.10}$

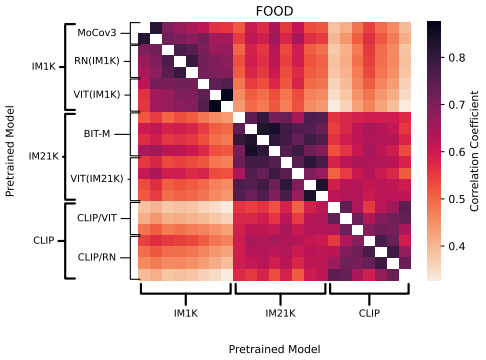


Figure 4. Pairwise correlation heatmap among any two pretrained models on FOOD dataset. Pretrained models trained with the same dataset tend to give similar information about neighborhood agreement, regardless of the architecture. IM1K: ImageNet-1K, IM21K: ImageNet-21K, indicating the pretraining dataset of the pretrained model. See Figure 10 for plots for each dataset.

for each pretrained model, which implies that a pretrained model with a potentially better performance in a downstream dataset can be more useful to detect failures for a model trained with this dataset.

Agreement Correlation Within and Between Model Families Beyond studying the effect of each pretrained model, we further investigate when two pretrained models tend to have similar agreement on the same sample. Specifically, we compute neighbor agreement between a trained model and a pretrained model for each sample. Hence, we obtain the pairwise correlation, i.e. Pearson Correlation, between any two pretrained models. Figure 4 and 10 display the overall correlation map and show an interesting observation: the models pretrained on the same datasets tend to give more similar information and are less affected by the architectures. This observation also implies that when pretrained models can achieve comparable performance, we should select the models that use different pre-training data to benefit from the more diverse information.

4.3. Failure Detection in OOD

We further verify our method under distribution shifts, which simulate the challenging tasks encountered in a real-world deployment. Classifiers may produce more inaccurate predictions when dealing with unseen data, highlighting the need for failure detection for safety.

The results are shown in Table 2, where we treat CIFAR10 and CIFAR100 as ID data and train classifiers based on these data, and test them on data distributions unseen during training: CIFAR10.1, corrupted CIFAR10 and CIFAR100. Similarly, we test on distribution shifts, ImageNetV2 and ImageNet-Sketch for ImageNet as ID data. All evaluation settings follow those used in ID data evaluation. The results show that our method can generalize well in distribution shift cases and generally outperform the baselines.

Table 2. AUROC (%) Performance on distribution shift datasets averaged on 6 runs. All evaluation settings follow those used in Table 1.

Model	Method	CIFAR10.1	CIFAR10-C	CIFAR100-C	ImageNetV2	ImageNet-SK
CNN	MSP	$89.09_{\pm 1.08}$	$77.83_{\pm 1.46}$	$77.70_{\pm 0.66}$	$83.93_{\pm 0.00}$	$79.40_{\pm 0.00}$
	Entropy	$89.11_{\pm 1.12}$	$78.03_{\pm 1.53}$	$78.41_{\pm 0.71}$	$81.63_{\pm 0.00}$	$\mathbf{80.01}_{\pm 0.00}$
	Energy	$86.94_{\pm 1.37}$	$77.20_{\pm 1.84}$	$79.18_{\pm 0.72}$	$71.24_{\pm 0.00}$	$75.77_{\pm 0.00}$
	MaxLogit	$87.09_{\pm 1.35}$	$77.40_{\pm 1.83}$	$79.38_{\pm 0.72}$	$75.37_{\pm 0.00}$	$77.56_{\pm 0.00}$
	TrustScore	$91.39_{\pm 0.19}$	$82.01_{\pm 0.77}$	$80.81_{\pm 0.57}$	$74.91_{\pm 1.01}$	$74.04_{\pm 1.26}$
	T.S.	$88.83_{\pm 1.18}$	$78.14_{\pm 1.58}$	$78.57_{\pm 0.81}$	$84.07_{\pm 0.00}$	$79.05_{\pm 0.02}$
	T.S. (w/ AS, single)	$95.20_{\pm 0.71}$	$88.96_{\pm 1.20}$	$80.64_{\pm 0.78}$	$84.11_{\pm 0.03}$	$79.16_{\pm 0.06}$
	T.S. (w/ AS, multiple)	$\mathbf{96.09}_{\pm 0.65}$	$\mathbf{92.76}_{\pm 0.53}$	$\mathbf{83.54}_{\pm 0.60}$	$\mathbf{84.11}_{\pm 0.04}$	$79.16_{\pm 0.10}$
ViT	MSP	$96.46_{\pm 0.27}$	$92.42_{\pm 0.62}$	$87.61_{\pm 0.53}$	$82.87_{\pm 0.00}$	$81.81_{\pm 0.00}$
	Entropy	$96.41_{\pm 0.28}$	$92.42_{\pm 0.62}$	$87.75_{\pm 0.49}$	$80.08_{\pm 0.00}$	$80.24_{\pm 0.00}$
	Energy	$91.39_{\pm 0.47}$	$89.19_{\pm 1.24}$	$83.39_{\pm 0.33}$	$66.39_{\pm 0.00}$	$72.43_{\pm 0.00}$
	MaxLogit	$91.69_{\pm 0.43}$	$89.42_{\pm 1.23}$	$83.96_{\pm 0.35}$	$73.61_{\pm 0.00}$	$77.05_{\pm 0.00}$
	TrustScore	$96.08_{\pm 0.36}$	$92.21_{\pm 0.77}$	$88.00_{\pm 0.47}$	$81.71_{\pm 0.43}$	$80.31_{\pm 0.31}$
	T.S.	$96.23_{\pm 0.29}$	$92.35_{\pm 0.66}$	$87.69_{\pm 0.48}$	$83.68_{\pm 0.02}$	$82.00_{\pm 0.00}$
	T.S. (w/ AS, single)	$96.28_{\pm 0.52}$	$92.61_{\pm 0.52}$	$87.87_{\pm 0.49}$	$84.38_{\pm 0.07}$	$83.00_{\pm 0.10}$
	T.S. (w/ AS, multiple)	$\mathbf{96.64}_{\pm 0.39}$	$\mathbf{93.48}_{\pm 0.33}$	$\mathbf{88.35}_{\pm 0.48}$	$\mathbf{84.77}_{\pm 0.06}$	$\mathbf{83.29}_{\pm 0.10}$

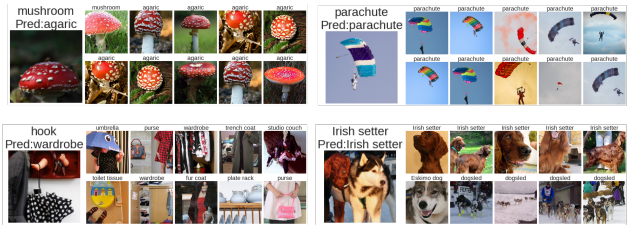


Figure 5. Four exemplar samples in ImageNet.top/bottom: samples with high/low agreement scores; left/right: failure/correct predictions. The top and bottom row of each sample box shows neighborhood samples under the trained model(fine-tuned ViT/B-16) and foundation model (i.e., CLIP ViT/L-14), respectively. The samples with lower agreement scores tend to have multiple objects in the pictures, leading to intrinsic difficulties for the model in correctly predicting and thus disagreement between models.

4.4. Case Study

What samples tend to get lower/higher agreement scores? Besides the numeric results, we also take a closer look at the samples that receive lower or higher agreement scores. Notably, as our method is based on neighborhood similarity, it enables some explanation ability by inspecting the difference between the neighborhood samples obtained from different models, especially when low agreement scores occur. We would like to highlight this as it enables human experts to further investigate the root cause of failed predictions.

Figure 5 shows the erroneous and correct samples with high and low agreement scores in ImageNet. We visualize the nearest 5 samples under the trained classifier (fine-tuned ViT/B-16) and a foundation model (CLIP ViT/L-14). We observe that samples with low agreement scores tend to be complex or contain multiple objects, leading to intrinsic difficulties for the model in correctly predicting and thus disagreement between models; while samples with high agreement scores usually contain a single prominent object.

4.5. Ablation Study

Effect of k and neighbor candidate pool size n In Figure 6, we analyze the effect of k and the neighbor candidate pool size n on two different datasets: CIFAR10 and CIFAR100 datasets, by varying the number of neighbors $k \in \{10, 20, 50, 100, 200, 500, 1000\}$ and neighbor candidate pool size $n \in \{2000, 5000, 10000, 20000, 50000\}$. It shows that the average performance is better and less sensitive to the choice k with increasing pool size n. In Table 7, we can see that even using a small subset of training data (e.g. $n = 2000$, $< 5\%$ training data from CIFAR10/CIFAR100) can already perform better than the baseline methods. Generally, the performance improves with larger pool sizes but stabilizes around an n of 10000 to 20000 for CIFAR10/CIFAR100.

Other choices of agreement measure To further study the impact of agreement measure, we replace NDCG with other similarity measures: Spearman's rank correlation coefficient, Centered Kernel Alignment (CKA) (Kornblith et al., 2019a) and Jaccard Similarity between k-hop neighborhood sets. For fair comparison, we also use k-hop neighborhood samples to compute CKA. The linear CKA and RBF-kernel CKA report similar results in our experiments. As shown in

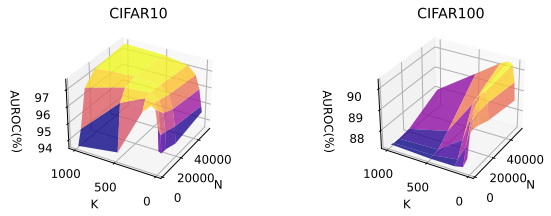


Figure 6. Ablation study with different k and n for CIFAR10 and CIFAR100. See the numeric results in Table 7.

Table 8, the results imply the importance of neighborhoods, adaptivity to more general transformations, and ranking information, by the comparison with Spearman, CKA, and Jaccard respectively. Note that, the adaptivity to more general transformations is most important among these factors as NDCG and Jaccard outperform CKA and Spearman, which might be credited to the fact that NDCG is invariant to more general transformations compared to CKA.

5. Theoretical Analysis

In this section, we discuss how foundation models correlate with the reliability of a trained model prediction by latent spaces agreement and justify the use of NDCG scores as latent spaces agreement.

Setup For analysis, we discuss the cases under a regression problem. Let the training dataset contains N samples, $D' = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ is the i-th input sample and $y^{(i)}$ is the ground-truth scalar value. We can denote the predictor $f_{w,B}(x) = \mathbf{w}^\top B(\mathbf{x})$, which consists of a feature extractor with normalization $B : \mathbb{R}^d \to \mathbb{R}^k$, and a weight vector $\mathbf{w} \in \mathbb{R}^{k\times 1}$. Thus, given the training dataset D', one can obtain a well-trained predictor $f_{w_0,B_0}(x)$ by minimizing a loss function l, i.e. $\mathbf{w}_0, B_0 = \arg\min_{\mathbf{w},B} l(\mathbf{x}, y, \mathbf{w}, B)$, where the loss function can be squared loss, etc. We use $\|\cdot\|$ as ℓ_2 norm. Let $H : \mathbb{R}^d \to \mathbb{R}^k$ be the pretrained encoder for a foundation model.

5.1. Relation between Prediction Error and Latent Space Agreement

We denote loss function $l(\mathbf{x}, y, B, \mathbf{w}) := \|\mathbf{w}^\top B(\mathbf{x}) - y\|$. Let $\mathbf{w}_0 := \arg\min_{\mathbf{w}} \frac{1}{n}\sum_x \left\|\mathbf{w}^\top B_0(\mathbf{x}) - y\right\|$. We assume there exists an isometric transformation $U_h \in \mathcal{U}$, where $\mathcal{U}$ contains all possible isometric transformations and $U_h := \arg\min_U \mathbb{E}\,\|B_0(X) - UH(X)\|$. We assume that there exists a head $\mathbf{w}_h \in \mathbb{R}^{k\times 1}$, where $\left\|\mathbf{w}_h^\top U_h^{-1} - \mathbf{w}_0^\top\right\| = \|\Delta\| \le C$. We use normalized features, i.e. $\|B_0(\mathbf{x})\| = 1$.

Proposition 5.1. *Given a test sample* $\mathbf{x}$ *and its ground-truth value* y*, the trained encoder* B_0 *and its linear head* $\mathbf{w}_0$*, we have a pretrained encoder* H *from a foundation model. If the pretrained encoder predicts correctly:* $l(\mathbf{x}, y, H, \mathbf{w}_h) = 0$*, the prediction error* $l(\mathbf{x}, y, B_0, \mathbf{w}_0) \le (C + \|\mathbf{w}_0\|) \cdot \|B_0(\mathbf{x}) - U_h H(\mathbf{x})\| + C$.

Proof. The detailed proof is relegated to Appendix D. □

In summary, if two latent spaces highly agree on a sample $\mathbf{x}$, i.e. $\|B_0(\mathbf{x}) - U_h H(\mathbf{x})\|$ close to 0, the model is more likely to predict accurately on $\mathbf{x}$. However, measuring this latent space agreement $\|B_0(\mathbf{x}) - U_h H(\mathbf{x})\|$ can be challenging as there exists an unknown isometric transformation U_h.

5.2. Local Approximation Isometry and NDCG

Next, we show that NDCG provides an effective measure of similarity between latent spaces irrespective of rotations or distortion. To show this, we first define a transformation f which approximately preserves distances around $\mathbf{x}$, as a δ-Local Approximation Isometry:

Assumption 5.2. (δ-Local Approximation Isometry). $\forall \mathbf{z} \in \mathcal{N}_k(\mathbf{x}), \exists \delta \ge 1, \frac{\|f(\mathbf{z}) - f(\mathbf{x})\|}{\|\mathbf{z} - \mathbf{x}\|} \in \left(\frac{1}{\delta}, \delta\right)$.

Intuitively, δ describes how 'approximately' the distances are preserved by f. For example, when $\delta = 1$, all the samples around $\mathbf{x}$ are strictly distance-preserving, so the neighborhood and ranking of neighborhood samples do not change. As δ increases, the neighborhood after applying f differs more from the original neighborhood, as does the ranking of neighborhood samples.

Proposition 5.3. *(Lower Bound of NDCG scores). Given an input sample* $\mathbf{x}$*,* Π^* *and* Π' *are permutations before and after a* δ*-local approximation isometric transformation* f*, we have* $\mathsf{NDCG}(\Pi^*, \Pi', r) \ge \frac{1}{\delta^2}$*, when* $r = 1/d(\cdot, \mathbf{x})$ *and* d *is a distance scoring function.*

Proof. The detailed proof is relegated to Appendix D. □

This shows that if a local approximate isometry exists near a point $\mathbf{x}$, the NDCG is guaranteed to be high ($\ge 1/\delta^2$). As δ approaches 1, the NDCG also approaches 1.

6. Conclusions

While powerful foundation models have received increasing attention, their use in improving model reliability is still underexplored due to the challenges of incompatible latent spaces between foundation models and a trained classifier. In this paper, we proposed a novel inter-model latent agreement framework to overcome this incompatible issue and improve the reliability of a trained classifier without any fine-tuning. We first show the agreement correlates well with classification accuracy. Motivated by this, our framework enables incorporating the agreement score into predictive confidence to improve failure detection performance.

We conduct extensive experiments on failure detection to verify the benefits of our framework to improve model reliability and provide theoretical justification for our method. We believe our proposed neighborhood agreement measure between latent spaces can further benefit the study of the interconnection between different models.

Acknowledgements

This research is supported by the National Research Foundation Singapore under its AI Singapore Programme (Award Number: [AISG2-TC-2021-002]). We thank Shen Li for proofreading and useful suggestions.

References

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9640–9649, October 2021.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Deng, A., Li, S., Xiong, M., Chen, Z., and Hooi, B. Trust, but verify: Using self-supervised probing to improve trustworthiness. In *European Conference on Computer Vision*, pp. 361–377. Springer, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Fort, S., Ren, J., and Lakshminarayanan, B. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:7068–7081, 2021.

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

Galil, I., Dabbah, M., and El-Yaniv, R. Which models are innately best at uncertainty estimation? *arXiv preprint arXiv:2206.02152*, 2022.

Galil, I., Dabbah, M., and El-Yaniv, R. What can we learn from the selective prediction and uncertainty estimation performance of 523 imagenet classifiers? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=p66AzKi6Xim.

Gneiting, T., Balabdaoui, F., and Raftery, A. E. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69 (2):243–268, 2007.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hein, M., Andriushchenko, M., and Bitterwolf, J. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41–50, 2019.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.

Jaeger, P. F., Lüth, C. T., Klein, L., and Bungert, T. J. A call to reflect on evaluation practices for failure detection in image classification. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=YnkGMIh0gvX.

Järvelin, K. and Kekäläinen, J. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

Jiang, H., Kim, B., Guan, M. Y., and Gupta, M. To trust or not to trust a classifier. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 5546–5557, 2018.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, pp. 491–507, 2020.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019a.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019b.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Kuleshov, V. and Deshpande, S. Calibrated and sharp uncertainties in deep learning via density estimation. In *International Conference on Machine Learning*, pp. 11683–11693. PMLR, 2022.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020.

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., and Lucic, M. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694, 2021.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do cifar-10 classifiers generalize to cifar-10? 2018. https://arxiv.org/abs/1806.00451.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Renggli, C., Pinto, A. S., Rimanic, L., Puigcerver, J., Riquelme, C., Zhang, C., and Lučić, M. Which model to transfer? finding the needle in the growing haystack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9205–9214, 2022.

Steiner, A. P., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your vit? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=4nPswr1KcP.

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.

Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=5hLP5JY9S2d.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019.

Wang, Y., Wang, L., Li, Y., He, D., Chen, W., and Liu, T.-Y. A theoretical analysis of ndcg ranking measures. In *Proceedings of the 26th annual conference on learning theory (COLT 2013)*, volume 8, pp. 6, 2013.

Wightman, R. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Xiong, M., Li, S., Feng, W., Deng, A., Zhang, J., and Hooi, B. Birds of a feather trust together: Knowing when to trust a classifier via adaptive neighborhood aggregation. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=p5V8P2J61u.

A. Experimental Setup

A.1. Datasets

We run experiments on six in-distribution datasets and five distribution shifts to evaluate the failure detection performance. For in-distribution, we use CIFAR10 (Krizhevsky et al.), CIFAR100, STL (Coates et al., 2011), BIRDS (Wah et al., 2011), FOOD (Bossard et al., 2014) and a large-scale dataset, ImageNet (ImageNet-1K) (Deng et al., 2009). For distribution shifts, we use CIFAR10.1 (Recht et al., 2018), Gaussian Blur Corrupted CIFAR10 samples (CIFAR10-C) (Hendrycks & Dietterich, 2019) with a severity level of 5 as natural and corruption distribution shift for CIFAR10. Similarly, we use Gaussian Blur Corrupted CIFAR100 samples (CIFAR100-C) as corruption distribution shift for CIFAR100. For ImageNet, we fine-tune on ImageNet and evaluate failure detection on distribution shifts: ImageNetV2 (Recht et al., 2019) and ImageNet-Sketch (ImageNet-SK) (Wang et al., 2019). See details about datasets and split settings in Table 3.

Table 3. Number of images per data set and associated splits

Datasets	Classes	Train Size	Val. Size	Test Size	Unlabeled Set Size
CIFAR10	10	50000	1000	9000	-
CIFAR100	100	50000	1000	9000	-
BIRDS	200	5994	2897	2897	-
STL	10	5000	4000	4000	100000
FOOD	102	75750	12625	12625	-
ImageNet	1000	1281167	10000	40000	-
CIFAR10-C	10	-	-	10000	-
CIFAR100-C	10	-	-	10000	-
CIFAR10.1	10	-	-	2000	-
ImageNetV2	1000	-	-	10000	-
ImageNet-Sketch	1000	-	-	50000	-

A.2. Base Models

We consider two common architectures: CNN-base (ResNet-50) and ViT-base (ViT-B/16) models as base classifiers across all datasets. To see if our method can still outperform on "pretrained and fine-tuned" models, all our base classifiers are initialized with a larger-scale data pretrained model and then fine-tuned. For all datasets except ImageNet, our base classifiers are trained with initializing with ResNet-50 architecture pretrained on ImageNet-1K examples and ViT/B-16 model pretrained on ImageNet-21K examples. For ImageNet, we use public fine-tuned models from PyTorch Image Models (Wightman, 2019), which are ImageNet-21K pretrained ResNetV2-50 model and CLIP pretrained ViT/B-16 model. We use penultimate layer output as the encoding feature vectors for the trained models.

Model Architectures and pretraining source For all datasets except for ImageNet, our base models are trained with initializing with ResNet-50 model pretrained on ImageNet-1K examples and ViT/B-16 model pretrained on ImageNet-21K examples. For ImageNet, we use public fine-tuned models from TIMM (Wightman, 2019), which are fine-tuned on ImageNet based on CLIP ViT/B-16 model.

Training Receipt For ResNet-50 models, we fine-tune with Adam optimizer with learning rate $1e-4$ and $(\beta_1, \beta_2) = (0.9, 0.99)$. For CIFAR10, CIFAR100, STL and BIRDS, we fine-tune for 50 epochs. For FOOD, we fine-tune for 20 epochs. We use the public trained ImageNet classifier from (Wightman, 2019).

For ViT, we fine-tuned with cosine annealing scheduler. The detail is shown in Table 4.

Base Models Performance For sanity check of trained classifiers, we show the average classification accuracy of our trained models used in this paper in Table 5.

Table 4. Training parameters per data set for ViT. init-lr: Initial learning rate of the cosine annealing scheduler as selected. steps: Number of batches that was trained on.

Datasets	init-lr	batch size	steps
CIFAR10	3e-4	64	15000
CIFAR100	3e-4	64	15000
BIRDS	3e-4	64	5000
STL	3e-4	64	4000
FOOD	3e-4	32	47000

Table 5. Average classification accuracy (%) of trained classifiers used in our paper.

Model	CIFAR10	CIFAR100	STL	BIRDS	FOOD	ImageNet
CNN	97.36	85.04	97.79	77.68	81.53	80.31
ViT	99.09	93.47	99.33	84.76	90.60	85.24

A.3. Foundation/Pretrained Models

In this work, we have included 23 public pretrained models of diverse architectures, training data and optimization losses. For specific, these models can be categorized into 5 model families as: CLIP ViT/ResNet (Radford et al., 2021), ViT (pretrained on ImageNet-21K or ImageNet-1K) (Dosovitskiy et al., 2020; Steiner et al., 2022), BiT-M (Kolesnikov et al., 2020), ResNet (He et al., 2016) and MoCov3 (Chen et al., 2021). Among the candidate models, the model with best fine-tune ImageNet accuracy is CLIP ViT/L-14 (87.85%) and the second best is ViT/L-16 (ImageNet-21K) (87.08%)[3]. We use penultimate layer output as the encoding feature vectors for the pretrained models. Except for multi-modal foundation model CLIP, we use the image encoders. We introduce the model families as follows:

- **CLIP-RN/VIT** We include four ResNet-based contrastive CLIP models (ResNet-50, ResNet-101, ResNet50x4, ResNet50x64) and three ViT-based CLIP models (ViT/B-32, ViT/B-16, ViT/L-14).
- **Vision Transformer (ViT)** We include ViT models pretrained on ImageNet-1K (Steiner et al., 2022; Dosovitskiy et al., 2020)(ViT/S-16, ViT/B-16, ViT/B-16@384px) and ImageNet-21K(ViT/T-16, ViT/S-16, ViT/B-16, ViT/L-16).
- **BIT-M** We use four ResNetv2-based model pretrained in ImageNet21K: ResNetv2-50, ResNetv2-50x3, ResNetv2-101, RseNetv2-152x4.
- **ResNet** We use three ResNet models prerained in ImageNet-1K: ResNet-50, ResNet-101, ResNet-152.
- **MoCov3** We include the self-supervised pretrained models with ResNet-50 and ViT/B-16 architectures.

A.4. Method Implementation

Hyperparameters We have training set size n and neighborhood size k as hyperparameters. For main results, except for the ablation study, we use $n = 10000$ across all datasets, except for BIRDS with 5994 training samples in total. The candidate pool is sampled from the training set except for STL, for which we sample from the unlabeled set. We select $k \in \{10, 20, 50, 100, 200, 500, 1000\}$ with optimal AUROC performance on validation split for each dataset. See Table 6.

we extract the features with different pretrained encoders and save for the later test stage. For feature extracting, we only require one-pass inference cost on training sample set, which is low computational compared to fully fine-tuning or adapting.

A.5. Baselines

Our baseline methods include the Maximum Softmax Probability (MSP) (Hendrycks & Gimpel, 2017), other uncertainty measure from the trained model: Entropy and Energy (Lakshminarayanan et al., 2017; Liu et al., 2020), the distance based measure: TrustScore (Jiang et al., 2018), MaxLogit proposed for distribution shift (Vaze et al., 2022) and the vanilla Temperature Scaling (T.S.) (Guo et al., 2017).

[3]as reported in (Wightman, 2019).

Table 6. Hyperparameters used in each dataset. Source: the data source of neighborhood samples. As STL has unlabeled samples, we sample from its unlabeled split as neighborhood sample sets. Evaluation in distribution shifts use the same setting as the corresponding ID.

Datasets	Source	n	k
CIFAR10	train	10000	200
CIFAR100	train	10000	20
BIRDS	train	5994	10
STL	unlabeled	10000	200
FOOD	train	10000	20
ImageNet	train	10000	10

A.6. Evaluation Metrics

Following the evaluation in (Hendrycks & Gimpel, 2017), we treat success/error prediction as positive and negative respectively, and use the area under the receiver operating characteristic curve (AUROC) as evaluation metric, with a bigger value indicating a more accurate failure detection.

B. Framework

Algorithm 1 Inter-model Latent Agreement

Input: Training dataset D_{tr}, validation dataset D_{val}, the encoder B from a trained classifier, the pretrained encoders $\mathcal{H} = \{H_1, \ldots, H_m\}$ from m foundation models, test sample $\mathbf{x}^{\text{test}}$
Output: the adjusted probabilities: $\tilde{P}(Y \mid \mathbf{x}^{\text{test}})$
Collect feature vectors with encoders B and $H_1, \ldots, H_m$ based on D_{tr} as $\mathbb{Z}$ and $\mathbb{H}^1, \ldots, \mathbb{H}^m$. ▷ Eq(4)(5)
Calibration:
For $\mathbf{x}$ in D_{val}, we compute $\mathsf{AS}(\mathbf{x}, B, \mathcal{H})$ ▷ Eq(9)
Obtain $\tau^*(\mathbf{x})$ by minimizing the NLL loss on D_{val} ▷ Eq(10)(11)
Test Stage:
Given a test sample $\mathbf{x}^{\text{test}}$, we compute $\mathsf{AS}(\mathbf{x}^{\text{test}}, B, \mathcal{H})$ ▷ Eq(9)
Return: the adjusted probabilities: $\tilde{P}(Y \mid \mathbf{x}^{\text{test}})$ with $\tau^*(\mathbf{x}^{\text{test}})$ ▷ Eq(11)

C. Failure Detection Results

C.1. Ablation Study

We show the numeric results of ablation study on hyperparameters and choices of agreement measures in Table 7 and 8, respectively.

Table 7. Ablation study on different k and N

	MSP	T.S.	N=2000	N=5000	N=10000	N=20000	N=50000
CIFAR10	94.15	93.76	97.26	97.40	97.45	97.53	97.50
CIFAR100	88.42	87.21	89.08	89.58	89.63	90.16	90.38

We can see that even using a small subset of training data (e.g. N=2000, $< 5\%$ training data from CIFAR10/CIFAR100) can already perform better than the baseline methods. The numeric results show that our method works well without a large amount of training data as the pool. Generally, the performance improves with larger pool sizes but stabilizes around an N of 10000 to 20000. Note that, with varying n, we select $k \in \{10, 20, 50, 100, 200, 500, 1000\}$ with optimal AUROC performance on validation split, which is following our main result experimental protocol.

We replace NDCG with other similarity measures: Spearman's rank correlation coefficient, Centered Kernel Alignment (CKA) (Kornblith et al., 2019a) and Jaccard Similarity between k-hop neighborhood sets. For fair comparison, we also use k-hop neighborhood samples to compute CKA. The linear CKA and RBF-kernel CKA report similar results in our

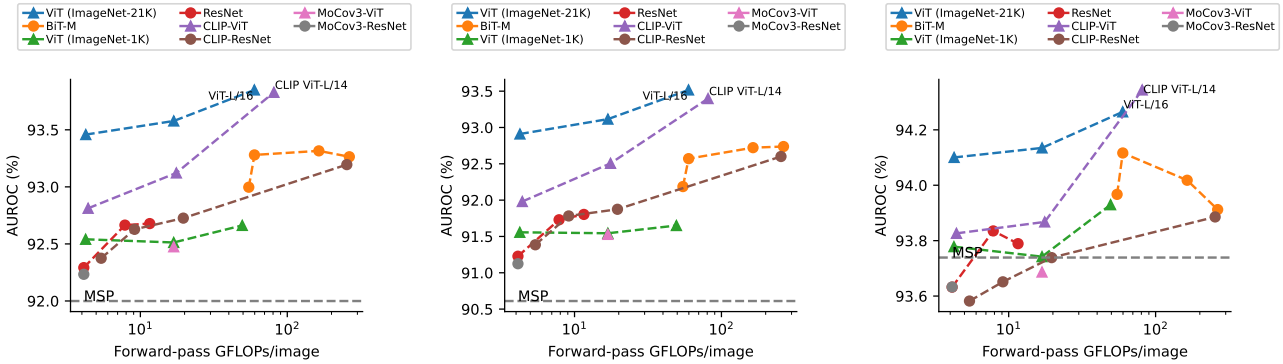
Table 8. Ablation study on different choices of agreement measures. The performance is averaged over all ID datasets.

Method	Avg AUROC(%)
MSP	91.11
T.S.	90.98
T.S. (w/ Spearmanr, single)	90.87
T.S. (w/ CKA, single)	90.99
T.S. (w/ Jaccard, single)	92.59
T.S. (w/ AS, single)	**92.67**

experiments. In Table 8, the results imply the importance of neighborhoods, adaptivity to more general transformations, and ranking information, by the comparison with Spearman, CKA, and Jaccard respectively. Note that, the adaptivity to more general transformations is most important among these factors as NDCG and Jaccard outperform CKA and Spearman, which might be credited to the fact that NDCG is invariant to more general transformations compared to CKA.

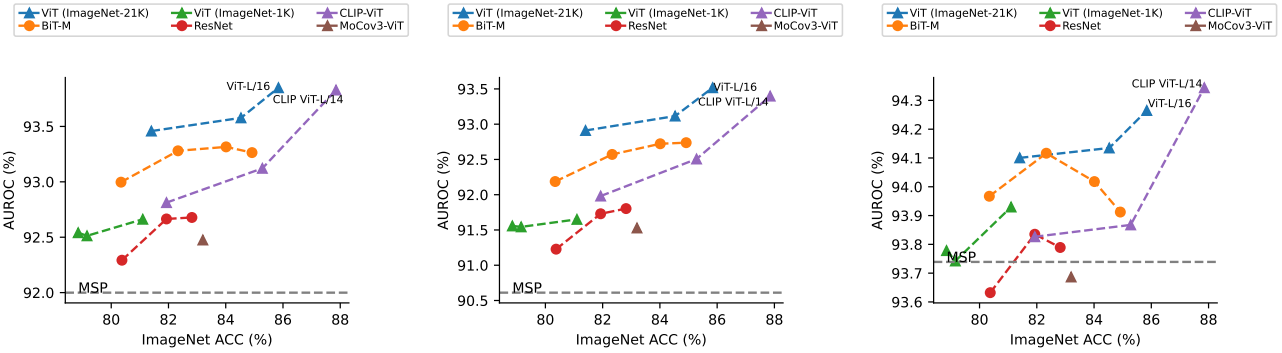
C.2. Exploration Study

Note that, the exploration study about KNN Accuracy and Correlation heatmap is conducted on CNN-based trained classifiers across different datasets.



(a) Average over CNN and ViT (b) CNN Classifiers (c) ViT Classifiers

Figure 7. Failure detection performance AUROC(%) for each pretrained model average over all ID datasets (ImageNet excluded). x-axis: inference GPU cost.



(a) Average over CNN and ViT (b) CNN Classifiers (c) ViT Classifiers

Figure 8. Failure detection performance AUROC(%) for each pretrained model average over all ID datasets (ImageNet excluded). x-axis: finetune ImageNet accuracy (%) of pretrained model.

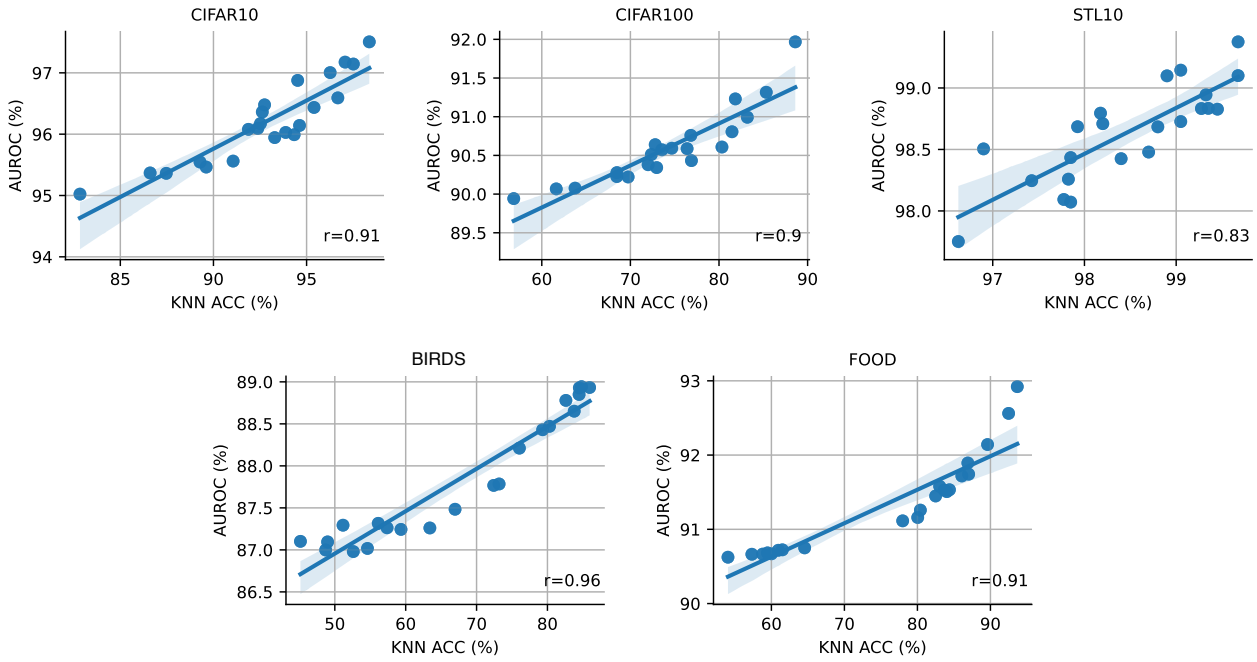


Figure 9. Strong correlation between Failure detection performance AUROC(%) for pretrained model and its KNN accuracy performance in each dataset.

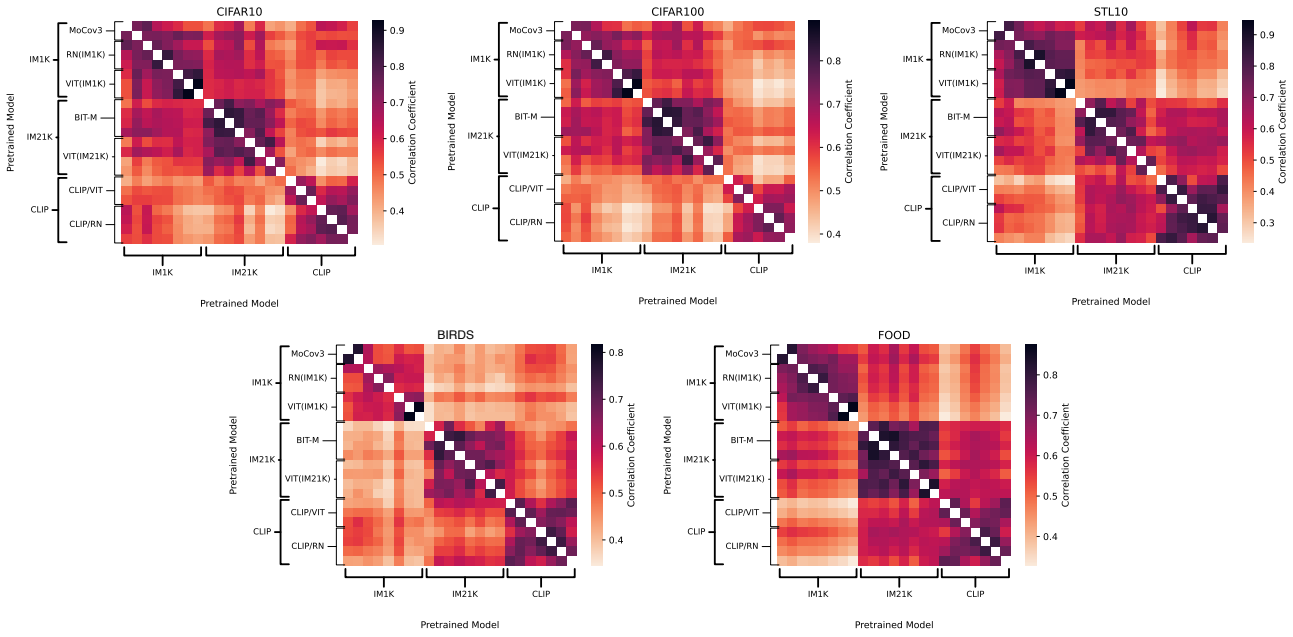


Figure 10. Correlation heatmap inter different pre-trained model families.

D. Proof of Theoretical Analysis

Setup we discuss the cases under a regression problem. Let the training dataset contains N samples, $D' = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$ is the i-th input sample and $y^{(i)}$ is the ground-truth scalar value. We can denote the predictor $f_{w,B}(x) = \mathbf{w}^\top B(\mathbf{x})$, which consists of a feature extractor with normalization $B : \mathbb{R}^d \to \mathbb{R}^k$, and a weight vector $\mathbf{w} \in \mathbb{R}^{k\times 1}$. Thus, given the training dataset D', one can obtain a well-trained predictor $f_{w_0,B_0}(x)$ by minimizing a loss function l, i.e. $\mathbf{w}_0, B_0 = \arg\min_{\mathbf{w},B} l(\mathbf{x}, y, \mathbf{w}, B)$, where the loss function can be squared loss, etc. We use $\|\cdot\|$ as ℓ_2 norm. Let $H : \mathbb{R}^d \to \mathbb{R}^k$ be the pretrained encoder for a foundation model.

We denote $l(\mathbf{x}, y, B, \mathbf{w}) := \|\mathbf{w}^\top B(\mathbf{x}) - y\|$. Let $\mathbf{w}_0 := \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{\mathbf{x}} \left\|\mathbf{w}^\top B_0(x) - y\right\|$. We assume there exists an rotation transformation $U_h \in \mathcal{U}$, where $\mathcal{U}$ contains all possible rotation transformations and $U_h := \arg\min_U \mathbb{E}\,\|B_0(X) - UH(X)\|$. We assume that there exists a head $\mathbf{w}_h \in \mathbb{R}^{k\times 1}$, where $\left\|\mathbf{w}_h^\top U_h^{-1} - \mathbf{w}_0^\top\right\| = \|\Delta\| \le C$. We use normalized features, i.e. $\|B_0(\mathbf{x})\| = 1$.

Proposition D.1. *Given a test sample* $\mathbf{x}$ *and its ground-truth value* y*, the trained encoder* B_0 *and its linear head* $\mathbf{w}_0$*, we have a pretrained encoder* H *from a foundation model. If the pretrained encoder predicts correctly:* $l(\mathbf{x}, y, H, \mathbf{w}_h) = 0$*, the prediction error* $l(\mathbf{x}, y, B_0, \mathbf{w}_0) \le (C + \|\mathbf{w}_0\|) \cdot \|B_0(\mathbf{x}) - U_h H(\mathbf{x})\| + C$.

Proof.

$$
\begin{aligned}
l(\mathbf{x}, y, B_0, \mathbf{w}_0) &= \left\|\mathbf{w}_0^\top B_0(\mathbf{x}) - y\right\| && (12)\\
&= \left\|(\mathbf{w}_h^\top U_h^{-1} - \Delta)(U_h H(\mathbf{x}) + B_0(\mathbf{x}) - U_h H(\mathbf{x})) - y\right\| && (13)\\
&= \left\|\mathbf{w}_h^\top U_h^{-1} U_h H(\mathbf{x}) - y + \mathbf{w}_h^\top U_h^{-1}(B_0(\mathbf{x}) - U_h H(\mathbf{x}))) - \Delta B_0(\mathbf{x})\right\| && (14)\\
&= \left\|\mathbf{w}_h^\top U_h^{-1}(B_0(\mathbf{x}) - U_h H(\mathbf{x})) - \Delta B_0(\mathbf{x})\right\| && (15)\\
&\overset{(a)}{\le} \left\|\mathbf{w}_h^\top\right\| \cdot \|B_0(\mathbf{x}) - U_h H(\mathbf{x})\| + C && (16)\\
&\le (C + \|\mathbf{w}_0\|) \cdot \|B_0(\mathbf{x}) - U_h H(\mathbf{x})\| + C, && (17)
\end{aligned}
$$

where (a) comes from $\|\Delta\| \le C$ and $\|U_h\| = 1$. □

In summary, that is if two latent spaces highly agree on a sample $\mathbf{x}$, i.e. $\|B_0(\mathbf{x}) - U_h H(\mathbf{x})\|$ close to 0, the model is more likely to predict accurately on $\mathbf{x}$.

Recall that if a transformation f can approximately preserve the distance around x after the transformation, we call this transformation δ-Local Approximation Isometry:

Assumption D.2. (δ-Local Approximation Isometry). $\forall \mathbf{z} \in \mathcal{N}_k(\mathbf{x}), \exists \delta \ge 1, \frac{\|f(\mathbf{z}) - f(\mathbf{x})\|}{\|\mathbf{z} - \mathbf{x}\|} \in \left(\frac{1}{\delta}, \delta\right)$.

Proposition D.3. *(Lower Bound of NDCG scores). Given an input sample* $\mathbf{x}$*,* Π^* *and* Π' *are permutations before and after a* δ*-local approximation isometric transformation* f*, we have* $\mathsf{NDCG}(\Pi^*, \Pi', r) \ge \frac{1}{\delta^2}$*, when* $r(\cdot) = 1/d(\cdot, \mathbf{x})$ *and* d *is a distance scoring function.*

Proof.

$$
\begin{aligned}
\mathsf{NDCG}(\Pi^*, \Pi', r) = \frac{\sum_i^n \frac{r(\Pi'_{(i)})}{\log(i+1)}}{\sum_i^n \frac{r(\Pi^*_{(i)})}{\log(i+1)}} &= \frac{\sum_i^n \frac{1/d(\mathbf{x}_{\Pi'_{(i)}}, \mathbf{x})}{\log(i+1)}}{\sum_i^n \frac{1/d(\mathbf{x}_{\Pi^*_{(i)}}, \mathbf{x})}{\log(i+1)}} && (18)\\
&\overset{(a)}{\ge} \frac{\sum_i^n \frac{1/d(f(\mathbf{x}_{\Pi'_{(i)}}), f(\mathbf{x}))}{\log(i+1)} \cdot \frac{1}{\delta}}{\sum_i^n \frac{1/d(f(\mathbf{x}_{\Pi^*_{(i)}}), f(\mathbf{x}))}{\log(i+1)} \cdot \delta} && (19)\\
&\overset{(b)}{=} \frac{\sum_i^n \frac{r'(\Pi'_{(i)})}{\log(i+1)} \cdot \frac{1}{\delta}}{\sum_i^n \frac{r'(\Pi^*_{(i)})}{\log(i+1)} \cdot \delta} && (20)\\
&\overset{(c)}{\ge} \frac{1}{\delta^2}, && (21)
\end{aligned}
$$

where $r'(\cdot) = 1/d(f(\cdot), f(\mathbf{x}))$, (a) comes from the definition of δ-local approximation isometry, (b) is substituting r' and (c) comes from the Rearrangement Inequality - note that in the numerator, the terms $r'(\Pi'_{(i)})$ are arranged in non-increasing order with i, since Π' is defined as the permutation which sorts samples based on their distances to x after applying the f function, i.e. the distances defining the score r'. As such, in the numerator, both $r'(\Pi'_{(i)})$ and $1/\log(i+1)$ are arranged in the same order: $r'(\Pi'_{(1)}) \geq \cdots \geq r'(\Pi'_{(n)})$ and $1/\log(i+1) \geq \cdots \geq 1/\log(n+1)$. In contrast, in the denominator, the same r' terms are present but in a (possibly) different order, so the denominator is smaller than or equal to the numerator by the Rearrangement Inequality. Equality is achieved when the two permutations, Π^* and Π', are the same.

□

Intuitively, it shows that when the mapping between two spaces f is a δ-local approximation isometric transformation, if δ approaches 1 (i.e. more distance-preserving / similar between the two spaces), the proposed metric based on NDCG is guaranteed to be high ($\geq 1/\delta^2$), implying that our score function accurately reflects how similar the two latent spaces are.