A The calculation of Dispersion Score

Our proposed approach, Dispersion Score, can be calculated as shown in Algorithm 1.

Algorithm 1 OOD Error Estimation via Dispersion Score

```
Input: OOD test dataset $\tilde{\mathcal{D}} = \{\tilde{x}_i\}_{i=1}^{m}$, a trained model $f$ (feature extractor $f_g$ and classifier $f_\omega$)
Output: The dispersion score
for each OOD instance $\tilde{x}_i$ do
    Obtain feature representation via $\tilde{\boldsymbol{z}}_i = f_g(\tilde{\boldsymbol{x}}_i)$.
    Obtain pseudo labels via $\tilde{y}_i' = \arg\max f_\omega(\boldsymbol{z}_i)$
end for
Calculate cluster centroids $\{\tilde{\boldsymbol{\mu}}_j\}_{j=1}^{k}$ using pseudo labels $\tilde{y}_i'$, with $\tilde{\boldsymbol{\mu}}_j = \frac{1}{m_j}\sum_{i=1}^{m_j} \boldsymbol{z}_i \cdot \mathbb{1}\{\tilde{y}_i' = j\}$
Calculate the feature center of all instances by $\bar{\boldsymbol{\mu}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{z}_i$
Calculate Dispersion Score $S(\tilde{\mathcal{D}})$ via Equation (4)
```

B Related work

Predicting generalization. Since the generalization capability of deep networks under distribution shifts is a mysterious desideratum, a surge of researches pay attention to estimate the generalization capability from two directions.

1) Some works aim to measure generalization gap between training and test accuracy with only training data [Corneanu et al., 2020, Jiang et al., 2019, Neyshabur et al., 2017, Unterthiner et al., 2020, Yak et al., 2019, Martin and Mahoney, 2020]. For example, the model-architecture-based method [Corneanu et al., 2020] summarizes the persistent topological map of a trained model to formulate its inner-working function, which represents the generalization gap. Margin distribution [Jiang et al., 2019] measures the gap by gauging the distance between training examples and the decision boundary. However, those methods are designed for the identical distribution between the training and test dataset, being vulnerable to distribution shift.

2) Some studies try to estimate generalization performance on a specific OOD test dataset without annotation during evaluation. Many of them utilize softmax outputs of the shifted test dataset to form a quantitative indicator of OOD error [Guillory et al., 2021, Jiang et al., 2021, Guillory et al., 2021, Garg et al., 2022]. However, those methods are unreliable across diverse distribution shifts due to the overconfidence problem [Wei et al., 2022]. Another popular direction considers the negative correlation between distribution difference and model's performance in the space of features [Deng and Zheng, 2021] or parameters [Yu et al., 2022]. Nevertheless, common distribution distances practically fail to induce stable error estimation under distribution shift [Guillory et al., 2021], and those methods are usually computationally expensive. Unsupervised loss such as agreement among multiple classifiers [Jiang et al., 2021, Madani et al., 2004, Platanios et al., 2016, 2017] and data augmentation [Deng et al., 2021] is also employed for OOD error prediction, which requires specific model structures during training. In this work, we focus on exploring the connection between feature separability and generalization performance under distribution shift, which is training-free and does not have extra requirements for datasets and model architectures.

Exploring Feature distribution in deep learning. In the literature, feature distribution has been widely studied in domain adaptation [Ben-David et al., 2006, Pan et al., 2010, Zhuang et al., 2015, Tzeng et al., 2017], representation learning [Bengio et al., 2013, HaoChen et al., 2021, Ming et al., 2023, Huang et al., 2021], OOD generalization [Li et al., 2018, Chen et al., 2021, Wang et al., 2021], and noisy-label learning [Zhu et al., 2021, 2022]. Domain adaptation methods usually learn a domain-invariant feature representation by narrowing the distribution distance between the two domains with certain criteria, such as maxinum mean discrepancy (MMD) [Pan et al., 2010], Kullback-Leibler (KL) divergence [Zhuang et al., 2015], central moment discrepancy (CMD) [Zellinger et al., 2017], and Wasserstein distance [Lee and Raginsky, 2017]. InfoNCE [Huang et al., 2021] shows a key factor of contrastive learning that the distance between class centers should be large enough. In learning with noisy labels, it has been shown that the feature clusterability can be used to estimate the transition matrix [Zhu et al., 2021]. To the best of our knowledge, we are the first to analyze the connection between feature separability and the final accuracy on OOD data.

C Sensitivity analysis: pseudo labels

Here, we conduct a sensitivity analysis by using ground-truth labels in our method. Table 5 illustrates that the performance with pseudo labels is comparable with the performance using ground-truth labels. This phenomenon is consistent with the previous method - ProjNorm [Yu et al., 2022], shown in Table 8 of their paper. The reason behind this could be that the trained model is capable of identifying certain semantic information from most corrupted examples, thereby retaining their representations in a cluster. Thus, the separability of feature clusters can serve as an indicator of the final prediction performance for corruption perturbations. Additionally, we provide a failure case of feature clusters separability for adversarial perturbations in Appendix F.

Table 5: Comparison of Dispersion Score with pseudo labels and true labels on CIFAR10, CIFAR100 and TinyImageNet. The best results are highlighted in **bold**.

Dataset	Network	Pseudo labels		True labels	
		R^2	ρ	R^2	ρ
CIFAR 10	ResNet18	0.968	**0.990**	**0.979**	0.989
	ResNet50	**0.987**	0.990	0.985	**0.991**
	WRN-50-2	**0.961**	**0.988**	0.945	0.987
	Average	**0.972**	**0.990**	0.970	0.989
CIFAR 100	ResNet18	**0.952**	0.988	0.915	**0.989**
	ResNet50	0.953	0.985	**0.959**	**0.989**
	WRN-50-2	**0.980**	0.991	0.978	**0.995**
	Average	**0.962**	0.988	0.950	**0.991**
TinyImageNet	ResNet18	**0.966**	**0.986**	0.937	0.985
	ResNet50	**0.977**	0.990	0.954	**0.995**
	WRN-50-2	0.968	0.986	**0.977**	**0.994**
	Average	**0.970**	0.987	0.956	**0.991**

D More results on class-imbalance settings

This section provides elaborated outcomes of training-free benchmarks under the setting of class imbalance, serving as a complement to the results presented in Table 3.

Table 6: Summary of prediction performance on **Imbalanced** CIFAR-10C and CIFAR-100C for training-free benchmarks, where R^2 refers to coefficients of determination, and ρ refers to Spearman correlation coefficients (higher is better)

Dataset	Network	Rotation		ConfScore		Entropy		AgreeScore		ATC		Fréchet	
		R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ
CIFAR 10	ResNet18	0.767	0.922	0.823	0.965	0.841	0.969	0.669	0.922	0.830	0.966	0.966	0.983
	ResNet50	0.787	0.946	0.870	0.975	0.887	0.977	0.765	0.953	0.883	0.975	0.916	0.975
	WRN-50-2	0.829	0.968	0.915	0.986	0.913	0.986	0.823	0.972	0.922	0.986	0.866	0.977
	Average	0.794	0.945	0.869	0.976	0.880	0.977	0.752	0.949	0.878	0.976	0.916	0.979
CIFAR 100	ResNet18	0.769	0.944	0.872	0.988	0.840	0.985	0.858	0.979	0.905	0.988	0.905	0.972
	ResNet50	0.847	0.964	0.875	0.986	0.826	0.978	0.832	0.973	0.880	0.986	0.855	0.979
	WRN-50-2	0.930	0.981	0.976	0.993	0.980	0.993	0.944	0.981	0.981	0.994	0.889	0.988
	Average	0.849	0.963	0.908	0.989	0.882	0.985	0.878	0.978	0.922	0.989	0.883	0.980

E Partial OOD error prediction

In previous experiments, a common assumption is that the test set contains instances from all classes. To further explore the flexibility of our method on OOD test set, we introduce a new setting called *partial OOD error prediction*, where the label space for the test set is a subset of the label space for the training data.

Here, we train ResNet18 and ResNet50 on both CIFAR-10 and CIFAR-100 with 10 and 100 categories, respectively. Different from previous settings, we evaluate the prediction performance on CIFAR-10C and CIFAR-100C with the first 50% of categories. The numerical results are shown in Tabel 7.

Table 7: Summary of prediction performance on **partial sets** of CIFAR-10C and CIFAR-100C, where R^2 refers to coefficients of determination, and ρ refers to Spearman correlation coefficients (higher is better). The best results are highlighted in **bold**.

Dataset	Network	Rotation		ConfScore		Entropy		AgreeScore		ATC		Frechet		ProjNorm		Ours	
		R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ
CIFAR 10	ResNet18	0.578	0.896	0.795	0.982	0.826	0.984	0.615	0.931	0.802	0.981	0.842	0.941	0.770	0.968	**0.935**	**0.985**
	ResNet50	0.719	0.939	0.885	**0.993**	0.892	**0.993**	0.787	0.976	0.887	**0.993**	0.757	0.953	0.856	0.967	**0.950**	0.992
	Average	0.649	0.918	0.841	0.987	0.859	**0.989**	0.701	0.954	0.845	0.987	0.800	0.947	0.813	0.968	**0.942**	0.988
CIFAR 100	ResNet18	0.876	0.946	0.922	0.985	0.902	0.980	0.904	0.971	**0.943**	**0.986**	0.894	0.972	0.770	0.968	0.935	0.985
	ResNet50	0.923	0.967	0.917	0.980	0.890	0.975	0.915	0.976	0.932	0.983	0.837	0.978	0.856	0.967	**0.950**	**0.992**
	Average	0.899	0.956	0.920	0.983	0.896	0.978	0.909	0.973	0.938	0.984	0.866	0.975	0.813	0.968	**0.942**	**0.989**

From the results, we could observe that our method is more robust than existing methods in the setting of partial OOD error prediction. For example, ProjNorm suffers from the incomplete test dataset during the self-training process, with a dramatic drop from around 0.950 to around 0.810 for R^2 of CIFAR-10C and CIFAR-100C on average. Contrastively, our method still achieves high accuracy in predicting OOD errors, maintaining an average R^2 value of 0.950.

F Adversarial vs. Corruption robustness.

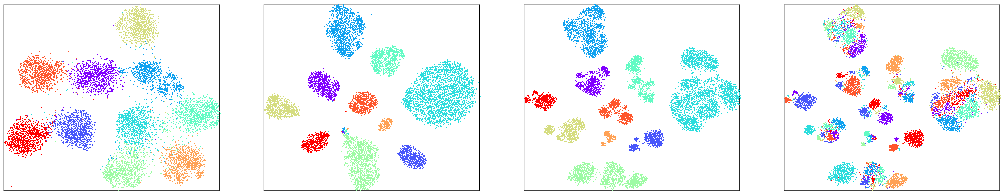
In previous analysis, we show the superior performance of dispersion score on predicting the accuracy on OOD test sets with different corruptions. Here, we surprisingly find that feature dispersion can effectively demonstrate the difference between adversarial and corruption robustness.

Table 8: Prediction performance measured by MSE against adversarial attack of different methods. The linear regression model is estimated on CIFAR-10C, and is used to predict the adversarial examples with perturbation size ϵ varying from 0.25 to 8.0. "True Dispersion" refers to the dispersion score with feature normalization using ground-truth labels. The best results are highlighted in **bold**.

	ConfScore	Entropy	ATC	ProjNorm	Dispersion	True Dispersion
CIFAR-10	0.933	0.892	0.906	0.847	1.359	**0.483**

Table 8 shows the prediction performance of different methods under adversarial attacks. "True Dispersion" refers to the dispersion score with feature normalization using ground-truth labels. In particular, we generate adversarial samples attacked by projected gradient descent (PGD) using untargeted attack [Kurakin et al., 2016] on the test set of CIFAR-10 with 10 steps and perturbation size ϵ ranging from 0.25 to 8.0. While the vanilla Dispersion score leads to poor performance, we note that the variant of Dispersion score with ground-truth labels performs much better than previous methods. This phenomenon is different from the conclusion of the sensitivity analysis of pseudo labels in predicting corruption robustness (See Appendix C), where the variant of true labels cannot outperform our method.

To understand the reasons behind the performance disparity of feature dispersion between adversarial and corruption robustness, we present the t-SNE visualization of features for adversarial attack of CIFAR-10 test set with various perturbation sizes in Figure 7. Compared to Figure 3, the results indicate that adversarial perturbations increase the distance between different clusters, whereas corruption perturbations decrease the separability of the clusters. In other words, adversarial perturbations decrease the test accuracy in a different way: assigning instances to the wrong groups and enlarging the distance among those groups. Therefore, feature dispersion using pseudo labels cannot be an effective method in the adversarial setting. We hope this insight can inspire specific designed methods based on feature dispersion for predicting adversarial errors in the future.



(a) ϵ=0.0 || pseudo (b) ϵ=2.0 || pseudo (c) ϵ=8.0 || pseudo (d) ϵ=8.0 || true

Figure 7: t-SNE visualization of feature representation on adversarial attack of CIFAR-10 test set with perturbation size ϵ ranging from 0.25 to 8.0.

G Performance on realistic datasets

To verify the effectiveness of Dispersion Score on realistic datasets, we conduct experiments on PACS [Li et al., 2017], Office-31 [Saenko et al., 2010] and Office-Home [Venkateswara et al., 2017] with ResNet-18, ResNet-50 and WRN-50-2 using normalization. Table 9 is the numerical results, from which we can observe that our method outperforms the other baselines on datasets with natural shifts.

Table 9: Performance comparison of all approaches on PACS, Office-31 and Office-Home, where R^2 refers to coefficients of determination, and ρ refers to Spearman correlation coefficients (higher is better). The best results are highlighted in **bold**.

Dataset	Network	Rotation		ConfScore		Entropy		AgreeScore		ATC		Fréchet		ProjNorm		Dispersion	
		R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ	R^2	ρ
PACS	ResNet18	0.823	**0.895**	0.595	0.755	0.624	0.755	0.624	0.832	0.514	0.650	0.624	0.804	0.161	0.420	**0.843**	0.846
	ResNet50	**0.861**	**0.923**	0.071	0.070	0.062	0.056	0.463	0.622	0.192	0.266	0.463	0.622	0.245	0.587	0.827	0.867
	WRN-50-2	0.865	0.902	0.646	0.678	0.629	0.671	0.377	0.858	0.753	0.832	0.558	0.832	0.475	0.650	**0.896**	**0.937**
	Average	0.850	0.907	0.437	0.501	0.438	**0.494**	0.488	0.771	0.486	0.583	0.549	0.753	0.294	0.552	**0.855**	**0.883**
Office-31	ResNet18	0.753	0.943	0.470	0.829	0.322	0.714	0.003	0.086	**0.844**	**0.943**	0.144	0.257	0.099	0.429	0.834	**0.943**
	ResNet50	0.371	0.829	0.486	0.829	0.355	0.829	0.012	0.464	0.533	0.486	0.035	0.257	0.241	0.429	**0.878**	**0.943**
	WRN-50-2	0.578	0.600	0.525	0.714	0.425	0.714	0.003	0.257	0.405	0.943	0.035	0.143	0.147	0.143	**0.798**	**0.829**
	Average	0.567	0.790	0.936	0.494	0.790	0.367	0.752	0.006	0.269	**0.594**	0.790	0.219	0.162	0.333	**0.836**	**0.905**
Office-Home	ResNet18	**0.823**	**0.930**	0.795	0.909	0.762	0.881	0.055	0.147	0.571	0.615	0.606	0.755	0.065	0.203	0.821	0.811
	ResNet50	**0.851**	**0.944**	0.770	0.895	0.742	0.853	0.027	0.217	0.487	0.734	0.607	0.685	0.169	0.476	0.841	0.860
	WRNt-50-2	0.823	0.958	0.742	0.874	0.696	0.846	0.132	0.406	0.384	0.643	0.589	0.706	0.173	0.531	**0.897**	**0.937**
	Average	0.832	0.944	0.769	0.893	0.734	0.860	0.071	0.256	0.481	0.664	0.601	0.716	0.135	0.403	**0.853**	**0.869**