Flatten Long-Range Loss Landscapes for Cross-Domain Few-Shot Learning

Yixiong Zou, Yicong Liu, Yiman Hu, Yuhua Li, Ruixuan Li*

School of Computer Science and Technology, Huazhong University of Science and Technology

{yixiongz, smnight, m202273659, idcliyuhua, rxli}@hust.edu.cn

Abstract

Cross-domain few-shot learning (CDFSL) aims to acquire knowledge from limited training data in the target domain by leveraging prior knowledge transferred from source domains with abundant training samples. CDFSL faces challenges in transferring knowledge across dissimilar domains and fine-tuning models with limited training data. To address these challenges, we initially extend the analysis of loss landscapes from the parameter space to the representation space, which allows us to simultaneously interpret the transferring and fine-tuning difficulties of CDFSL models. We observe that sharp minima in the loss landscapes of the representation space result in representations that are hard to transfer and fine-tune. Moreover, existing flatness-based methods have limited generalization ability due to their short-range flatness. To enhance the transferability and facilitate fine-tuning, we introduce a simple yet effective approach to achieve long-range flattening of the minima in the loss landscape. This approach considers representations that are differently normalized as minima in the loss landscape and flattens the high-loss region in the middle by randomly sampling interpolated representations. We implement this method as a new normalization layer that replaces the original one in both CNNs and ViTs. This layer is simple and lightweight, introducing only a minimal number of additional parameters. Experimental results on 8 datasets demonstrate that our approach outperforms state-of-the-art methods in terms of average accuracy. Moreover, our method achieves performance improvements of up to 9% compared to the current best approaches on individual datasets. Our code will be released.

1. Introduction

Cross-domain few-shot learning (CDFSL) is introduced as a solution to mitigate the need for extensive training data on the target domain. It leverages knowledge transferred from non-overlapping source domain datasets that have sufficient training data [2, 11]. It poses two primary chal-
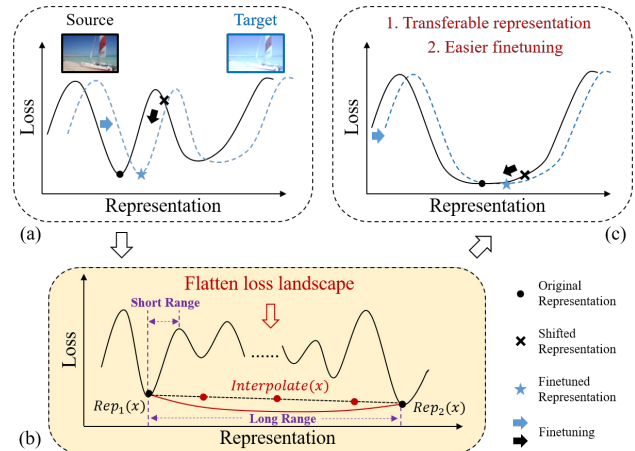
---

*Corresponding author. Code is at https://github.com/Zoilsen/FLoR.

Figure 1. (a) Representation-space loss landscape (RSLL): Given an input sample, the model maps it into a representation space, where effective representations correspond to low classification losses, i.e., minima in the landscape. Since domain shifts can be reflected by the landscape shift and representation shift, a sharp minimum in the landscape corresponds to a representation vulnerable to domain shifts, making the training and finetuning difficult. (b) We can easily find different minima in the RSLL, which covers a longer range than current flatness-based methods. This inspires us to flatten a long-range loss landscape by randomly interpolating these minima. (c) Given a flattened minimum, the representation and the model are more transferable against domain shifts and easier to be finetuned on the target domain.

lenges due to domain shifts between the source and target domains. (1) **Transferring**: overcoming domain shifts to transfer knowledge from source to target datasets. (2) **Fine-tuning**: leveraging the transferred knowledge to learn from limited training data in the target domain. Numerous approaches [20, 36] have been proposed to address these challenges. However, these challenges still remain unsolved.

Currently, several works [8, 18, 34] have attempted to address the challenge posed by domain shifts by investigating the flatness of the loss landscape. However, these works primarily concentrate on the parameter space, which can not reflect the presence of domain shifts directly. To tackle the CDFSL problem, we extend this concept from the parameter space to the representation space, which provides a more

direct means of capturing domain shifts.

In the source domain training phase, a model is trained to extract effective representations for each input training sample. The representation for a source-domain sample can be seen as a point in the representation space, where good representations correspond to low classification losses. Consequently, we can draw landscapes to map each representation point to its classification loss, with effective representations corresponding to minima, as illustrated in Fig. 1a. Note that this sample can also be represented by its original pixels, which enables the direct analysis of domain shifts.

Since the model is trained on the source domain dataset, the representation extracted by this model is effective enough for recognition. Therefore, each source-domain representation (in feature space or pixel space) extracted by this model corresponds to a minimum in the loss landscape. By applying domain shifts to this source-domain sample, its representation may shift from the original minimum (black dot in Fig. 1) to another point (black cross in Fig. 1). Compare Fig. 1a and Fig. 1c, intuitively, we can see (1) a flat minimum would tolerate larger representation shifts, making the transferring easier; and (2) a flat minimum would reduce the high-loss region between the source and target representations, making the finetuning easier.

Therefore, our goal is to flatten the loss landscape in the representation space during source-domain training to enhance both transferring and fine-tuning. However, we find that the current flatness-based method [8] has limited performance on the CDFSL task. By experiments, we conclude this is because current methods rely on the flatness of the landscape in the vicinity of the minimum, which covers only a short range. Therefore, it struggles to effectively address large domain shifts in CDFSL.

To handle this problem, we notice that **we can easily find different minima in the representation space**: given an un-normalized input representation, many normalization methods have been proved to be effective in producing discriminative normalized outputs, which can be viewed as different minima in the representation space. As the distance between representations is much larger than the sunken region in the loss landscape around each minimum, we can flatten a long-range loss landscape between these minima.

Based on this intuition, we randomly interpolate differently normalized representations at each layer of deep networks, and use interpolated representations for classification. This would push interpolated representations to be effective, lowering the high-loss region between minima and flattening the loss landscape. Specifically, we provide two instantiations for both Convolutional Neural Networks and Vision Transformers, and implement the above operations as a normalization layer (FLoR layer) to replace the original normalization layer. Our contribution can be listed as

• To the best of our knowledge, we are the first to extend the analysis of loss landscapes from the parameter space to the representation space for the CDFSL task, which interprets the difficulty in transferring and finetuning.

• Based on the analysis, we propose a simple but effective method to flatten the loss landscape in a long range, which simultaneously enhances the transferring and finetuning of the model.

• We evaluate our model on 8 datasets to show the effectiveness and rationale, indicating we can outperform state-of-the-art works in the average accuracy, and outperform current best works on individual datasets by up to 9%.

2. Analyzing Generalization from the Aspect of Representation-Space Loss Landscapes

2.1. Preliminaries

Cross-domain few-shot learning (CDFSL) aims to recognize target-domain novel classes (C^{novel}) by only a few training samples, with knowledge transferred from source-domain base classes (C^{base}) with sufficient training data [11]. Note that $C^{novel} \cap C^{base} = \varnothing$, and a domain gap exists between these two sets of classes [11]. The model is firstly trained on the base-class dataset $D^{base} = \{x_i, y_i\}_i^N$ where $y_i \in C^{base}$ (base-class stage). Typically, the base-class training is to minimize the cross-entropy loss

$$L = L_{cls}(F(x_i), y_i), \qquad (1)$$

where $F(x_i) = h(f(x_i))$ outputs the classification probability of x_i, which is composed of a feature extractor $f(\cdot)$ and a classifier $h(\cdot)$. Then, $f(\cdot)$ is transferred to novel-class dataset $D^{novel} = \{x_i^u, y_i^u\}_i^{N^u}$ where $y_i^u \in C^{novel}$ (novel-class stage). Typically, only 1 or 5 training samples are available for each novel class, which makes the training on novel classes difficult. For a fair comparison, during the novel-class stage, current works [11] always sample k-way n-shot episodes for novel-class training and evaluation. Each episode contains a support set $S = \{x_{ij}^s, y_{ij}^s\}_{i,j}^{k,n}$ with k novel classes and n training samples in each class for training, and a query set $Q = \{x_i^q\}_i^{N^q}$ for evaluation. Denote the model trained on novel classes as $F^u(\cdot) = h^u(f^u(\cdot))$, the prediction for x_i^q is

$$\hat{y}_i^u = \arg max F^u(x_i^u), \qquad (2)$$

Finally, the evaluation can be conducted based on the comparison between $\hat{y}_i^u$ and the real label. As can be seen, two major tasks are in CFDSL: (1) obtaining a generalizable feature extractor on source-domain base classes, and (2) effectively finetuning the model given only a few samples on target-domain novel classes. To handle these two tasks, we first analyze model's robustness to domain shifts, by expanding the concept of loss-landscape flatness from the parameter space to the representation space.

2.2. Representation-Space Loss Landscape

Given a deep network, its Parameter-Space Loss Landscape (PSLL) [8, 18, 34] refers to drawing a curve to map its pa-

rameters to classification losses. When the data changes, the model parameters trained on this data will also change, leading to a shift in the loss landscape. However, as the data shift is reflected in the parameters trained on the data, it is indirect to see how the domain shift influences the model decision, since the model training is not deterministic. To handle this problem, we try to directly analyze how the shift in pixels or representations would influence the model decision (Representation-Space Loss Landscape, RSLL).

Specifically, denote a data point as x, it can be represented in the pixel space (e.g., RGB) or the feature space through a feature extractor, which we uniformly term as the representation space. Denote its representation as $r(x) \in R^{c \times h \times w}$. Since each point in the representation space can be mapped to a loss value, we can draw a landscape corresponding to each representation point and its loss value (black curve in Fig. 1a). During the base-class stage, the model is encouraged to learn a good representation that corresponds to a low classification loss, which can be understood as a minimum in the landscape (black dot in Fig. 1a). After training, each representation extracted by the model from each training sample can be viewed as a minimum.

Then, we fix and transfer this model to another data point x^u with domain shifts against x. To simplify the analysis, suppose x^u is directly generated by only shifting the domain information of x (Fig. 1a, the case of different images will be included later). Ideally, the model should also map x^u to the same representation point as the original data x, since ideally, the model is robust to domain shift. However, x^u is likely to be mapped to another point (black cross in Fig. 1a, $r(x^u)$). Since the model is fixed, the loss landscape is still the original one. Therefore, the mapped point $r(x^u)$ may correspond to a high loss, which means the representation $r(x^u)$ is not effective enough. If the original representation $r(x)$ is located in a sharp minimum, the shifted representation $r(x^u)$ will tend to be located in a high-loss region, demonstrating a badly transferred model. Therefore, we can flatten the landscape around the minimum in RSLL. **It would make the shifted representation located in the same low-loss region as the non-shifted representation, facilitating the model in tolerating larger representation shifts and making the transferring easier (Fig. 1c).**

If we finetune the model on the target domain, the model will be guided to minimize the loss of the representation, which will shift both the loss landscape (move the black curve to the blue curve in Fig. 1a) and the representation point[1] (move the black cross to the blue star in Fig. 1a) to obtain a new minimum (blue star in Fig. 1a). However, if the original representation $r(x)$ is located in a sharp minimum, there may be high-loss regions between $r(x)$ and $r(x^u)$. This would make it hard to move $r(x^u)$ to a low-loss region, especially for the few-shot scenarios, making the

---

[1]If the representation is in the pixel space, this point will not move.

variance = 0.0   variance = 0.1   variance = 0.2   variance = 0.3   variance = 0.4
(a) Pixel space with random domain perturbations

(b) Performance drop vs.
Image Perturbation variance

(c) Performance drop vs.
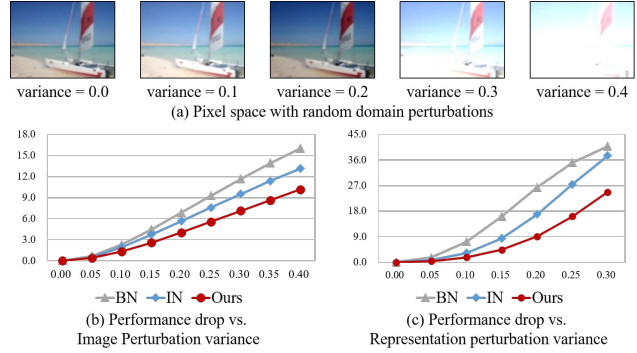Representation perturbation variance

Figure 2. To validate the analysis based on representation-space loss landscapes (RSLL), we apply low-frequency noises to the representation space (i.e., pixels and features) to be the domain shifts on training data for a sanity check. The perturbation variance measures the distance between the perturbed representation and the original representation (a minimum in RSLL). We use the performance drop against perturbation variance to measure the sharpness of the landscapes around the minimum, where a larger drop indicates a sharper minimum. We can see the model based on Instance Normalization (IN) is located in a flatter minimum than the model based on Batch Normalization (BN), which brings the high performance of the IN-based model in cross-domain tasks. This result is consistent with current works [10] and validates the rationale of the RSLL analysis. (a) Samples of the pixel perturbation. (b) Perturbation on pixels. (c) Perturbation on features.

few-shot finetuning difficult. Therefore, we can also flatten the landscape around the minimum in RSLL. **It would reduce the high-loss region between the source and target representations, making the finetuning easier (Fig. 1c).**

For the case of different images, there are two gaps between x and x^u: (1) the domain shift and (2) the semantic shift. Ideally, we can construct an intermediate sample x^I to represent x^u in the same domain of x but preserve the semantics of x^u. If the source domain training set of x is large enough, the model (e.g., Large Vision Models) trained on it will also ensure a good representation for x^I. Therefore, this case will come back to the case we analyzed above.

In summary, the benefits of RSLL analysis are in
- It handles domain shifts more directly than PSLL.
- It interprets the difficulty of transferring and finetuning.
- We can easily find many minima in RSLL (shown later).

2.3. Verification and Interpretation

To verify the rationale of the above analysis, we first experiment with the influence of domain shifts in the representation space for a sanity check. Since the Batch Normalization [17] is frequently applied in the widely adopted backbone network for CDFSL, and Instance Normalization [26] has been validated to be beneficial for cross-domain representations [10], we compare these the network with these two networks to validate the rationale of the above analysis. Due to the difficulty in plotting the loss landscape in the high dimensional representation space, we perturb the rep-
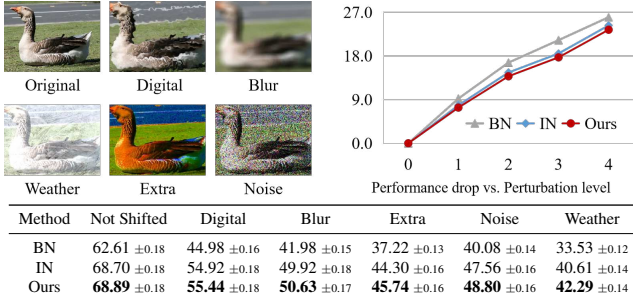
Method	Not Shifted	Digital	Blur	Extra	Noise	Weather
BN	62.61 ±0.18	44.98 ±0.16	41.98 ±0.15	37.22 ±0.13	40.08 ±0.14	33.53 ±0.12
IN	68.70 ±0.18	54.92 ±0.18	49.92 ±0.18	44.30 ±0.16	47.56 ±0.16	40.61 ±0.14
Ours	**68.89** ±0.18	**55.44** ±0.18	**50.63** ±0.17	**45.74** ±0.16	**48.80** ±0.16	**42.29** ±0.14

Table 1. To verify our analysis also holds for the real-world data, we evaluate our model on the test data of domain shifted *mini*ImageNet by 5-way 1-shot accuracy. Datasets are sampled from the ImageNet-c dataset with 5 kinds of shifts. This dataset provides 5 levels of perturbations, so we use the perturbation level to measure how the representation shifts from the original representation. The sharpness is also measured by the performance drop. We can see IN-based model still shows a flatter minimum in RSLL, which leads to its high performance in these datasets.
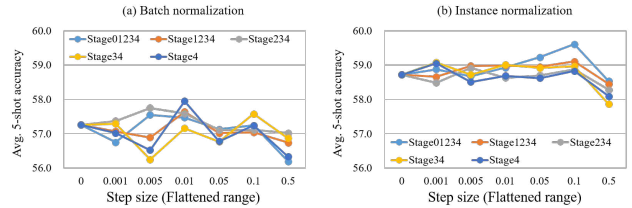

Figure 3. Directly apply SAM on each representation of ResNet10 on CDFSL datasets. Only marginal improvements on the average CDFSL 5-way 5-shot accuracy can be observed, and the perturbation step size is small, indicating the complex loss landscape could only support SAM to learn a short-range flatness.

Table 2. Our perturbation step size (flattened range) is much larger than SAM in the representation output by each layer, no matter when it is at the beginning (Init.) or the end of the training.

Rep. ID	1	2	3	4	5	6	7	8	9	10	11	12
Init.	5.06	3.45	2.12	3.58	1.99	3.57	3.25	2.33	3.34	4.69	3.80	4.84
End.	2.91	2.44	1.51	2.60	2.09	1.94	3.54	3.05	1.80	5.69	12.46	9.18

resentation of each data to see the performance drop against the perturbation variance. This variance measures the distance between the perturbed representation and the original representation (a minimum in RSLL). A larger drop indicates a sharper minimum in the landscape.

We first apply pixel perturbation (i.e., representing each data point by RGB pixels) to the training data. Current works [9] show that the domain information is always correlated with the low-frequency information in the Fourier-transformed images. Therefore, we randomly sample a low-frequency perturbation from the Gaussian Distribution, where the Variance controls the distance between the original image and the perturbed image. The perturbation is at the shape of $h^p \times w^p$ where h^p and w^p are much smaller than the image size to ensure the frequency is low. Then, the perturbation will be resized to the size of each image and finally applied to them. Examples of such perturbation are shown in Fig. 2a, and the magnitude of the performance drop is shown in Fig. 2b. As can be seen, the performance of all models drops under the perturbations, but the IN model drops less than the BN model, indicating its robustness to the domain shift. This result is consistent with current works [10], and verifies the IN model makes training images located in a flatter minimum in RSLL

Then, we apply perturbations to feature maps (i.e., represent each sample in the feature space). The perturbation is also the low-frequency Gaussian noise, and the performance drop is in Fig. 2c, where we can see similar results as in Fig. 2b. This result verifies the RSLL analysis also fits the feature map representation for each input sample.

To verify this analysis applies to real-world domain shifts, we then construct 5 domain-shifted *mini*ImageNet by selecting the same classes from ImageNet-c [14]. The domain shifts contain blur, digital, extra, noise, and weather

perturbations. Examples and results can be found in Tab. 1. This dataset provides 5 levels of perturbations for each kind of perturbation, so we utilize this level to represent the distance between the perturbed representation and the original one (a minimum in the RSLL). Similar to Fig. 2, the sharpness is measured by the performance drop. We can see the results are consistent with the toy experiments in Fig. 2, which further validates the rationale of the RSLL analysis.

In summary, by comparing IN and BN models on domain-shifted datasets, experiments in Fig. 2 and Tab. 1 validate (1) the RSLL analysis is rationale; (2) the domain perturbation on pixels and features, which we uniformly name as representation, are consistent. Moreover, it also interprets why IN models are better, as IN models' representations are in flatter minima compared with BN models.

2.4. Short-Range Flatness Limits Generalization

To improve the model's robustness to domain shifts, current works [8] provided an effective method named Sharpness-Aware Minimization (SAM). It applies adversarial perturbation to the model's parameters to flatten the PSLL. It inspires us to conduct SAM on the RSLL. Take the last layer's representation as an example, the training loss is

$$L = L_{cls}(h(f(x) + \eta \frac{\nabla f(x)}{||\nabla f(x)||}), y), \quad (3)$$

where η is the step size to control the flattened range, x is the input sample, y is the label, and $\nabla f(x) = \frac{\partial L}{\partial f(x)}$. Results are reported in Fig. 3, including the step size η and perturbed layers. However, we can only see marginal improvements against the baseline performance ($\eta = 0$).

Moreover, we can observe the step sizes are small, but the IN-based model could tolerate a larger step size with a higher performance. Combining this result with the results in Fig. 2 and Tab. 1, we hold that this is because the IN-based model extracts features in a flatter RSLL minimum.
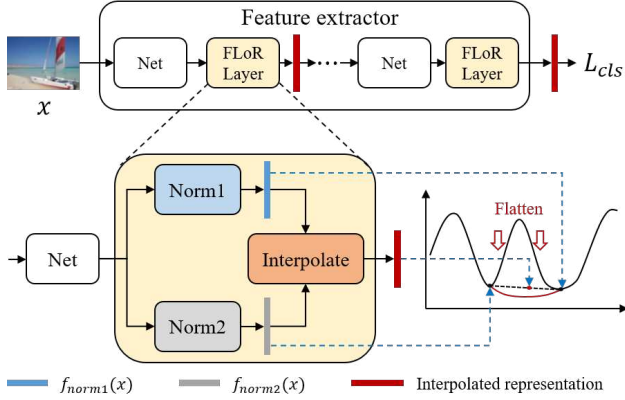
Figure 4. We implement our method as a normalization layer (FLoR layer) to replace the ordinary normalization layer in the backbone network (e.g., CNNs or ViTs). This layer interpolates two differently normalized representations, which flattens the intermediate high-loss region between two minima in RSLL.

As the perturbation direction is obtained by $\nabla f(x_i)$ which only estimates the local shape of RSLL, when the minimum is sharp, it fails to estimate the complex loss landscape shape outside the local minimum of $f(x_i)$. Therefore, the perturbation direction $\nabla f(x_i)$ will be less effective when the step size is large. In contrast, since IN-based representations are in flatter minima than BN-based ones, it ensures $\nabla f(x_i)$ is still effective given a larger perturbation step size. In this scenario, a larger step size would help the model to flat a longer range of RSLL and improve generalization.

Therefore, to learn a model robust to domain shifts, we aim to flatten a long-range RSLL around each representation. Our flattened range is reported in Tab. 2 on the representation output by each layer, where the range is much larger than ordinary SAM methods in Fig. 3.

3. Flatten Long-Range Loss Landscapes

To enlarge the range of flattened area, basically relying on the local information (such as gradient [8] or curvature [34]) of the optimum may not be enough, because this information is only effective inside the sunken area in the landscape.

To handle this problem, we notice a distinguished difference between the PSLL and RSLL: **there are many easy-to-find minima in RSLL**. For example, previous works proposed many different normalization methods, such as Batch Normalization or Instance Normalization. These methods can provide effective representations suitable for classification, although they may show differences in the fitness for different tasks. In other words, given the same training sample, different normalization methods could provide different minima in the same representation-space loss landscape. This inspires us to flatten the loss in the area between these minima (Fig. 1b).

Specifically, we utilize two normalization methods to normalize each pre-normalized representation, which pro-

duces two effective representations and corresponds to two minima in RSLL. Then, we propose to flatten the loss landscapes between these two minima, by means of classifying the input sample through the interpolated representations between two minima. Take the final layer's representation $f(x)$ as an example, the classification can be represented as

$$L = L_{cls}(h((1 - \delta)f_{norm1}(x) + \delta f_{norm2}(x)), y), \quad (4)$$

where $f_{norm1}(x)$ and $f_{norm2}(x)$ refer to two normalized representations, and $\delta \in [0, 1]$ is a ratio randomly sampled from the Beta distribution. In implementation, we conduct the interpolation in **every** layer's representation. Since the classification is based on the interpolated representation, this representation will be pushed to be effective, so it will be mapped to a low loss in the RSLL. Thus, the high-loss region between $f_{norm1}(x)$ and $f_{norm2}(x)$ will be flattened.

As this method is agnostic to the shape of the complex loss landscape between minima, we do not need to consider any local information around minima. We measure the distance between the BN and IN representations in Tab. 2, where the distance of each representation is much larger than step sizes in Fig. 3, indicating a larger flattened region.

Below, we provide two instantiations of the above idea for both Convolutional Neural Networks (CNN) [12] and Vision Transformers (ViT) [7] respectively.

3.1. Flattening for Convolutional Neural Networks

In CNNs, Batch Normalization is one of the most widely used normalization methods, which is typically applied after the convolution layer. BN refers to normalizing representations with batch statistics, which can be represented as

$$f_{BN}(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \quad (5)$$

$$\mu = \frac{1}{bhw} \sum_{i,j,k} X_{i,:,j,k}, \quad \sigma^2 = \frac{1}{bhw} \sum_{i,j,k} (X_{i,:,j,k} - \mu)^2,$$

where $X \in R^{b \times c \times h \times w}$ is the batch representation.

Instance normalization (IN) is another widely used normalization method for CNNs, which refers to normalizing representations with only the given image statistics as

$$f_{IN}(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \quad (6)$$

$$\mu = \frac{1}{hw} \sum_{j,k} X_{:,:,j,k}, \sigma^2 = \frac{1}{hw} \sum_{j,k} (X_{:,:,j,k} - \mu)^2.$$

For CNN, we set f_{norm1} to f_{BN}, and set f_{norm2} to f_{IN}.

3.2. Flattening for Vision Transformers

In ViTs, Layer Normalization is widely applied as the normalization method, which is represented as

$$f_{LN}(X) = \gamma \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \quad (7)$$

$$\mu = \frac{1}{c} \sum_{k} X_{:,:,k}, \quad \sigma^2 = \frac{1}{c} \sum_{:,:,k} (X_{:,:,k} - \mu)^2,$$

Table 3. Dataset information. Please see the appendix for details.

Dataset	Domain	Classes	Images
*mini*ImageNet	General recognition	64	38,400
CUB	Fine-grained bird recognition	50	2,953
Cars	Fine-grained car recognition	49	2,027
Plantae	Plantae recognition	50	17,253
Places	Scene recognition	19	3,800
CropDiseases	Agricultural disease recognition	38	43,456
EuroSAT	Satellite imagery recognition	10	27,000
ISIC2018	Skin lesion recognition	7	10,015
ChestX	X-ray chest recognition	7	25,847

where $X \in R^{b \times t \times c}$ is the batch representation.

Then, we follow [3] to only process the CLS token. We use BN as the second normalization method. In other words, we set f_{norm1} to f_{LN}, and set f_{norm2} to f_{BN} for the CLS token in each layer.

3.3. Implementation

In implementation, as shown in Fig. 4, we implement the above design as a normalization layer (FLoR layer) to replace the ordinary normalization layer in the backbone network. This layer can be represented as

$$f_{FLL}(x) = (1 - \delta)f_{norm1}(x) + \delta f_{norm2}(x), \quad (8)$$

which is the same as Eq. 4. Since δ is a random number sampled from the Beta distribution $Beta(a, b)$, our model imports only two hyper-parameters (a, b) and learnable parameters only in the added normalization layer, which is simple and lightweight.

During the base-class training, δ is randomly sampled from the $Beta(a, b)$. During the novel-class finetuning (if needed), δ is set to a learnable parameter that is finetuned together with the whole network, where its value is initialized as the expectation value of $Beta(a, b)$. Finally, we evaluate the model by either the prototype-based classifier or the finetune-based classifier, as illustrated in Eq. 2.

4. Experiments

4.1. Dataset and evaluation setup

Following current works [2, 11], our model is firstly trained on the *mini*ImageNet dataset [28] (base classes, source domain) and then transferred to 8 cross-domain dataset (novel classes, target domain, including CUB [29], Cars [19], Plantae [15], Places [35], CropDiseases [22], EuroSAT [13], ISIC2018 [5] and ChestX [31]) for few-shot training and evaluation, using the k-way n-shot classification. Dataset information is listed in Tab. 3.

4.2. Implementation details

During the base-class training, our model is trained with the AdamW optimizer [21] for 400 epochs with a learning rate of 0.001. For a fair comparison, we follow current works [10] to utilize ResNet10 [12] and Vit-S [7] (with DINO [33] pretraining on ImageNet1K [6]) as the backbone network. We set the Beta distribution as $a = b = 0.01$ for all experiments. During the novel-class finetuning, we use
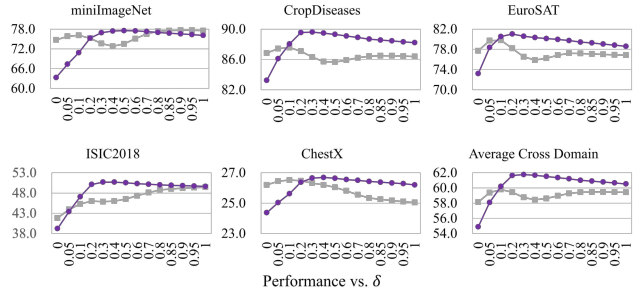

Figure 5. To verify the loss landscape between BN and IN representations, we train a baseline model with separate streams of IN and BN, by means of specifying different mixing ratios δ. Two peaks and one valley can be observed in the baseline curve (gray), indicating the high-loss region between the two representations. In contrast, only one peak is observed in our curve (purple), indicating we can effectively flatten the loss landscapes.

the SGD optimizer with a momentum of 0.9 to finetune all parameters. Please see the appendix for details.

4.3. Comparison with state-of-the-art works

Comparisons utilizing ResNet10 are listed in Tab. 4 and 5. For fairness, we group work by whether finetuning (FT) or the transductive (TR) setting is used. For all groups, we follow MN to use the cosine similarity, and subtract the mean of the support (query) set for inductive (transductive) calibration. For the transductive setting, we pseudo-label the query set by the prototype-based classifier or the finetune-based classifier, and use the pseudo-labeled query set to further update the classifier until convergence. As shown in Tab. 4 and 5, we achieve the top average performance in all settings, and achieve the highest performance in almost all datasets. Notably, on ISIC2018, we can significantly outperform state-of-the-art works by as much as 9% in the 5-shot setting. Comparisons with works using ViT-S are in Tab. 6, and we also achieve state-of-the-art performance.

4.4. Ablation study

4.4.1 Verification of local loss landscapes

To verify the loss landscape between two normalized representations, intermediate points are needed to be sampled between these two representations. We first train a network with a shared convolution layer and two separate normalizations in each convolution-normalization block. During training, each branch's final output will be classified separately. During the evaluation, we manually mix two representations in each layer of this model by specifying different δ in Eq. 8. Since the loss value is not consistent for comparison between different models, we measure the 5-way 5-shot accuracy of the plain prototype-based classification results on different domains, which is plotted in Fig. 5 as the gray curves. We can see that almost all datasets show two peaks and one valley in accuracy, verifying the existence of the high-loss region between the two representations.

Table 4. Comparison with state-of-the-art works based on ResNet10 by 5-way 1-shot accuracy.

Method	FT	TR	Mark	CUB	Cars	Places	Plantae	ChestX	ISIC2018	EuroSAT	CropDiseases	Average
GNN+FT [25]	×	×	ICLR-20	45.50	32.25	53.44	32.56	22.00	30.22	55.53	60.74	51.53
GNN+ATA [30]	×	×	IJCAI-21	45.00	33.61	53.57	34.42	22.10	33.21	61.35	67.47	43.84
MN+AFA [16]	×	×	ECCV-22	41.02	33.52	**54.66**	37.60	22.11	32.32	61.28	60.71	42.90
GNN+AFA [16]	×	×	ECCV-22	46.86	34.25	54.04	36.76	22.92	33.21	63.12	67.61	44.85
LDP-net [36]	×	×	CVPR-23	49.82	35.51	53.82	39.84	23.01	33.97	**65.11**	69.64	46.34
FLoR	×	×	**Ours**	**49.99**	**37.41**	53.18	**40.10**	**23.11**	**38.11**	62.90	**73.64**	**47.31**
Fine-tuning [11]	✓	×	ECCV-20	43.53	35.12	50.57	38.77	22.13	34.60	66.17	73.43	45.54
FLoR	✓	×	**Ours**	**50.01**	**38.13**	**53.61**	**40.20**	**23.12**	**38.81**	**69.13**	**84.04**	**49.63**
TPN+ATA [30]	×	✓	IJCAI-21	50.26	34.18	57.03	39.83	21.67	34.70	65.94	77.82	47.68
TPN+AFA [16]	×	✓	ECCV-22	50.85	38.43	60.29	40.27	21.69	34.25	66.17	72.44	48.05
FLoR	×	✓	**Ours**	**55.35**	**38.86**	**60.94**	**41.61**	**22.92**	**39.78**	**70.96**	**85.95**	**52.04**
TPN+ATA [30]	✓	✓	IJCAI-21	51.89	38.07	57.26	40.75	22.45	35.55	70.84	82.47	49.91
RDC [20]	✓	✓	CVPR-22	50.09	39.94	61.17	41.30	22.32	36.28	70.51	85.79	50.81
LDP-net [36]	✓	✓	CVPR-23	55.94	37.44	**62.21**	41.04	22.21	33.44	**73.25**	81.24	50.85
FLoR	✓	✓	**Ours**	**55.94**	**40.01**	61.27	**41.70**	**23.12**	**41.67**	71.38	**86.30**	**52.67**

Table 5. Comparison with state-of-the-art works based on ResNet10 by 5-way 5-shot accuracy.

Method	FT	TR	Mark	CUB	Cars	Places	Plantae	ChestX	ISIC2018	EuroSAT	CropDiseases	Average
GNN+FT [25]	×	×	ICLR-20	64.97	46.19	70.70	49.66	24.28	40.87	78.02	87.07	57.72
GNN+ATA [30]	×	×	IJCAI-21	66.22	49.14	75.48	52.69	24.32	44.91	83.75	90.59	60.89
MN+AFA [16]	×	×	ECCV-22	59.46	46.13	68.87	52.43	23.18	39.88	69.63	80.07	54.96
GNN+AFA [16]	×	×	ECCV-22	68.25	49.28	**76.21**	54.26	25.02	46.01	**85.58**	88.06	61.58
LDP-net [36]	×	×	CVPR-23	70.39	52.84	72.90	**58.49**	26.67	48.06	82.01	89.40	62.60
FLoR	×	×	**Ours**	70.39	**53.43**	72.31	55.80	**26.70**	**51.44**	80.87	**91.25**	**62.77**
Fine-tuning [11]	✓	×	ECCV-20	63.76	51.21	70.68	56.45	25.37	49.51	81.59	89.84	61.05
FLoR	✓	×	**Ours**	**73.39**	**57.21**	**72.37**	**61.11**	**26.77**	**56.74**	**83.06**	**92.33**	**65.37**
TPN+ATA [30]	×	✓	IJCAI-21	65.31	46.95	72.12	55.08	23.60	45.83	79.47	88.15	59.56
TPN+AFA [16]	×	✓	ECCV-22	65.86	47.89	72.81	55.67	23.47	46.29	80.12	85.69	59.73
FLoR	×	✓	**Ours**	**70.83**	**53.55**	**73.88**	**56.28**	**26.27**	**52.16**	**82.04**	**92.32**	**63.42**
TPN+ATA [30]	✓	✓	IJCAI-21	70.14	55.23	73.87	59.02	24.74	49.83	**85.47**	93.56	63.98
RDC [20]	✓	✓	CVPR-22	67.23	53.49	74.91	57.47	25.07	49.91	84.29	93.30	63.21
LDP-net [36]	✓	✓	CVPR-23	73.34	53.06	**75.47**	59.64	26.88	48.44	84.05	91.89	64.10
FLoR	✓	✓	**Ours**	**74.06**	**57.98**	74.25	**61.70**	**26.89**	**57.54**	83.76	**93.60**	**66.22**

Similarly, we also manually specify different δ to form a mixed representation for our model, and measure the corresponding accuracy (purple curve in Fig. 5). We can see (1) our model achieves higher performance, especially on cross-domain datasets, (2) there is only one peak in the curve and (3) the high-accuracy (low-loss) region is much larger than that of the baseline model, verifying the flatness.

4.4.2 Verification of each design

We ablate each design in Tab. 7 with the prototype classifier for fairness (i.e., fixing the backbone network, using Euclidean distance without domain calibration). We can see:

(1) **SAM on RSLL is better than PSLL**. We implement the sharpness-aware minimization (SAM [8]) on both the RSLL and PSLL, and see the performance on RSLL is higher, because RSLL handles domain shifts directly.

(2) **IN can partly account for our improvements**. By simply replacing BN with IN, the performance increases, since IN leads to a flatter minimum in the representation-space loss landscapes (verified in Fig. 2 and 1). But this is still much lower than ours.

(3) **Our perturbation direction is effective**. Since the flattening can be viewed to perturb representations, we compare it with some perturbation methods. We first follow Fig. 2 to add low-frequency noise for domain shifting on the input image (IN + Inp. PTB) and representations (IN + Rep.

PTB). The result is higher than that of IN but lower than ours, verifying the effectiveness of low-frequency noise and experiments in Fig. 2. An alternative is to perturb in the gradient direction (adversarial attack [1]), therefore we tried this option as IN + Inp Adv, whose performance is similar, due to the attacked semantics in samples

(4) **Learnable mixing ratio is not effective enough**. Some works also combine the output of BN and IN outputs, generating a learnable mixing ratio. Meta BIN [4] is chosen as a representative but its performance is also limited. This is because the learnable ratio simplifies the representation learning, while FLoR increases the difficulty as it additionally constrains the high-loss region between two minima.

(5) **Random ratio is better**. We test with two setups: Fix. BN + IN means to manually specify a pre-defined δ for both the base-class training and novel-class evaluation, and Sep. BN + IN refers to the model in Fig. 5. We can see such simple designs can lead to better results than Meta BIN which utilizes a learnable weight. However, since it can only cover one intermediate point while ours can randomly cover all points, our performance is better.

4.4.3 Verification of easier finetuning

We report the finetuning performance in Fig. 6. Typical finetuning methods [11, 37] refer to finetuning the backbone with a small learning rate and fixing shallow layers due to

Table 6. Comparison with state-of-the-art works with VIT-S pretrained by DINO on ImageNet-1K.

Method	Shot	FT	Mark	CUB	Cars	Places	Plantae	ChestX	ISIC2018	EuroSAT	CropDiseases	Average
StyleAdv [10]	1	×	CVPR-23	84.01	40.48	72.64	**55.52**	22.92	33.05	72.15	81.22	57.75
FLoR	1	×	**Ours**	**84.60**	**40.71**	**73.85**	51.93	22.78	**34.20**	**72.39**	**81.81**	**57.78**
PMF [23]	1	✓	CVPR-22	78.13	37.24	71.11	53.60	21.73	30.36	70.74	80.79	55.46
StyleAdv [10]	1	✓	CVPR-23	84.01	40.48	72.64	**55.52**	22.92	33.99	**74.93**	**84.11**	58.57
FLoR	1	✓	**Ours**	**85.40**	**43.42**	**74.69**	52.29	**23.26**	**35.49**	73.09	83.55	**58.90**
StyleAdv [10]	5	×	CVPR-23	95.82	61.73	88.33	**75.55**	**26.97**	47.73	88.57	94.85	72.44
FLoR	5	×	**Ours**	**96.18**	**61.75**	**89.23**	72.80	26.71	**49.52**	**90.41**	**95.28**	**72.74**
PMF [23]	5	✓	CVPR-22	-	-	-	-	**27.27**	50.12	85.98	92.96	-
StyleAdv [10]	5	✓	CVPR-23	95.82	66.02	88.33	**78.01**	26.97	51.23	90.12	95.99	74.06
FLoR	5	✓	**Ours**	**96.53**	**68.44**	**89.48**	76.22	27.02	**53.06**	**90.75**	**96.47**	**74.75**

Table 7. Ablation study of RSLL, Mixup, IN, perturbations and δ by 5-way 5-shot accuracy with plain prototype classifier.

Method	CropDisease	EuroSAT	ISIC2018	ChestX	Ave.
BN (Baseline)	85.80 ±0.27	78.01 ±0.22	39.10 ±0.33	26.13 ±0.17	57.26 ±0.13
PSLL SAM	86.60 ±0.44	76.83 ±0.50	40.42 ±0.42	26.03 ±0.34	57.47 ±0.38
RSLL SAM	86.19 ±0.32	78.03 ±0.33	41.26 ±0.31	26.26 ±0.17	57.95 ±0.17
Mixup	86.78 ±0.33	78.49 ±0.31	42.89 ±0.35	25.68 ±0.19	58.46 ±0.15
Manifold Mixup	87.04 ±0.34	78.90 ±0.33	42.16 ±0.30	26.01 ±0.16	58.53 ±0.15
IN	86.67 ±0.20	76.17 ±0.24	47.25 ±0.21	24.79 ±0.15	58.72 ±0.10
IN + Inp. PTB.	86.51 ±0.22	77.91 ±0.24	46.50 ±0.20	24.75 ±0.14	58.92 ±0.11
IN + Inp. Adv.	88.04 ±0.21	79.04 ±0.24	44.76 ±0.21	25.92 ±0.15	59.23 ±0.12
IN + Rep. PTB.	87.76 ±0.21	77.70 ±0.25	47.92 ±0.23	25.25 ±0.18	59.66 ±0.10
Meta BIN	86.66 ±0.19	78.52 ±0.28	46.06 ±0.31	25.28 ±0.16	59.13 ±0.12
Fix. BN + IN	86.66 ±0.21	77.79 ±0.27	48.67 ±0.29	25.26 ±0.17	59.59 ±0.13
Sep. BN + IN	88.51 ±0.20	79.38 ±0.19	44.85 ±0.21	26.25 ±0.18	59.66 ±0.14
Ours	**89.35** ±0.17	**79.40** ±0.27	**50.75** ±0.30	**26.57** ±0.16	**61.52** ±0.12

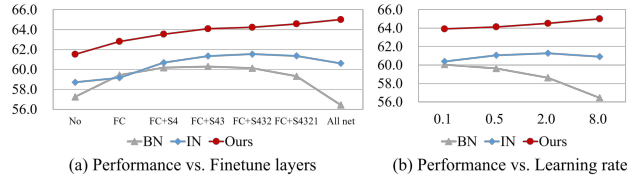(a) Performance vs. Finetune layers   (b) Performance vs. Learning rate

Figure 6. To verify the easy finetuning, we evaluate the finetuning with different layers and learning rates. FC: fully-connected layer, S: ResNet Stage, No: Prototype classifier without finetuning. Our model can directly finetune all layers with a large learning rate, indicating the easiness of few-shot finetuning.

the overfitting caused by scarce training data. We first test this claim by finetuning the BN baseline model (gray curve in Fig. 6). We can see this claim holds for this BN model. The best performance is achieved by fixing the first 3 stages and setting the learning rate to 0.1 times of the classifier, making the finetuning sensitive to hyper-parameter choices. In contrast, the IN model shows more layers can be finetuned with a larger learning rate. For ours, we can directly apply all layers into the finetuning, with a large learning rate at 8 times that in the classifier. This means our model is easier to finetune under the few-shot scenarios and is not sensitive to the hyper-parameter choices, which is the consequence of the flat minimum as validated in Fig. 2 and 5.

5. Related Work

5.1. Cross domain few-shot learning (CDFSL)

CDFSL aims to effectively learn from target domains with only a few training samples [2, 11]. Compared with the ordinary few-shot learning task, CDFSL needs to handle a larger domain gap, which makes it more difficult to transfer and finetune. Current works can be grouped into transferring-based [16, 30, 36] and finetuning-based works [11, 20]. Compared with them, our method focuses on source-domain training, but we can benefit the knowledge transfer and target-domain finetuning at the same time.

Some works [25, 36] show that the BN module is vulnerable to domain shifts. We also follow this conclusion, but we are the first to study the loss landscape around BN representations. Mixup [27, 32] is another line of learning

transferable models. Compared with them, our method only shares a similar formulation in interpolation but differs in (1) we do not need multiple samples for mixing, (2) we aim to flatten the loss landscape between two normalized representations, and (3) our results are much higher (Tab. 7).

5.2. Generalization based on loss landscapes

Current works show the relationship between loss landscapes and the generalization [8], where a widely adopted conclusion is that a flat minimum in loss landscapes refers to a more generalizable model. Most of these works analyze the loss landscapes from the **parameter space** [18] to handle the problem such as domain generalization [34] and catastrophic forgetting [24]. Compared with them, we are the first to introduce the analysis from the aspect of **representation-space** loss landscape into the CDFSL task.

6. Conclusion

We analyze model generalization from representation-space loss landscapes for CDFSL, and propose to flatten the long-range loss landscapes by randomly interpolating the differently normalized representations. Experiments on 8 datasets validate the effectiveness and rationale.

Acknowledgments

References

[1] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Adversarial attacks and defences: A survey, 2018. 7

[2] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *Proceedings of the International Conference on Learning Representations*, 2018. 1, 6, 8

[3] Xiaohui Chen, Yinkai Wang, Yuanqi Du, Soha Hassoun, and Li-Ping Liu. On separate normalization in self-supervised transformers, 2023. 6

[4] Seokeon Choi, Taekyung Kim, Minki Jeong, Hyoungseob Park, and Changick Kim. Meta batch-instance normalization for generalizable person re-identification, 2021. 7

[5] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic), 2019. 6

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 6

[7] 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, 2021. 5, 6

[8] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization, 2021. 1, 2, 4, 5, 7, 8

[9] Yuqian Fu, Yu Xie, Yanwei Fu, Jingjing Chen, and Yu-Gang Jiang. Wave-san: Wavelet based style augmentation network for cross-domain few-shot learning, 2022. 4

[10] Yuqian Fu, Yu Xie, Yanwei Fu, and Yu-Gang Jiang. Styleadv: Meta style adversarial training for cross-domain few-shot learning, 2023. 3, 4, 6, 8

[11] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pages 124–141. Springer, 2020. 1, 2, 6, 7, 8

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5, 6

[13] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification, 2019. 6

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019. 4

[15] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset, 2018. 6

[16] Yanxu Hu and Andy J. Ma. Adversarial feature augmentation for cross-domain few-shot classification, 2022. 7, 8

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456. PMLR, 2015. 3

[18] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima, 2017. 1, 2, 8

[19] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 6

[20] Pan Li, Shaogang Gong, Chengjie Wang, and Yanwei Fu. Ranking distance calibration for cross-domain few-shot learning, 2022. 1, 7, 8

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 6

[22] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7(September), 2016. Publisher Copyright: © 2016 Mohanty, Hughes and Salathé. 6

[23] Jan Stühmer Shell Xu, Da Li. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference, 2022. 8

[24] Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima, 2021. 8

[25] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *Proceedings of the International Conference on Learning Representations*, 2020. 7, 8

[26] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization, 2017. 3

[27] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states, 2019. 8

[28] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 3637–3645, 2016. 6

[29] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 6

[30] Haoqing Wang and Zhi-Hong Deng. Cross-domain few-shot classification via adversarial task augmentation, 2021. 7, 8

[31] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mo-hammadhadi Bagheri, and Ronald M. Summers. ChestX-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. 6

[32] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. 8

[33] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022. 6

[34] Xingxuan Zhang, Renzhe Xu, Han Yu, Hao Zou, and Peng Cui. Gradient norm aware minimization seeks first-order flatness and improves generalization, 2023. 1, 2, 5, 8

[35] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 6

[36] Fei Zhou, Peng Wang, Lei Zhang, Wei Wei, and Yanning Zhang. Revisiting prototypical network for cross domain few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20061–20070, 2023. 1, 7, 8

[37] Yixiong Zou, Shanghang Zhang, Yuhua Li, and Ruixuan Li. Margin-based few-shot class-incremental learning with class-level overfitting mitigation. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022. 7