Appendix

The appendix provides more details about the main paper in both methods in Sec.A and experiments in Sec.B.

A More Details about SFDA

A.1 Interpretation of weighted Kendall's tau

The Kendall's τ represents the ratio of concordant pairs minus discordant pairs when enumerating all $\binom{M}{2}$ pairs of $\{T_m\}_{m=1}^M$ and $\{G_m\}_{m=1}^M$ as given by

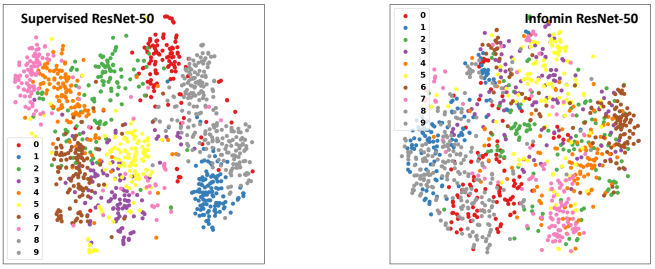
$$\tau = \frac{2}{M(M-1)} \sum_{1 \le i < j \le M} \text{sgn}(G_i - G_j)\text{sgn}(T_i - T_j) \tag{10}$$

where $\text{sgn}(x)$ is a sign function returning 1 if $x > 0$ and -1 otherwise. Moreover, we use a weighted version of Kendall's τ, denoted as τ_w, to evaluate transferability metrics considering that a best performing pre-trained model is always preferred for target task in transfer learning. τ_w can measure the ranking performance of top performing models. In principle, a larger τ_w indicates the transferability metric can produce a better ranking for pre-trained models.

A.2 Algorithm of SFDA

Algorithm 1 Pipeline of SFDA.

Input: target dataset $\mathcal{T} = \{(x_n, y_n)\}_{n=1}^N$; M pre-trained models $\{\phi_m\}_{m=1}^M$;
Hyper-parameters: $a = 4$ in Eqn.(3) to generate the regularization coefficient.
Output: the transferability scores of SFDA, $\{T_m\}_{m=1}^M$;.
Set: self-challenge = True
for $m = 1$ **to** M **do**
    calculate: static representations $\hat{x} = \theta_m(x)$.
    split: $\mathcal{T} = \{\hat{x}_n^{(1)}\}_{n=1}^{N_1} \cup \cdots \cup \{\hat{x}_n^{(C)}\}_{n=1}^{N_C}$.
    calculate: $\mu = \sum_{n=1}^{N} \hat{x}_n, \mu_c = \sum_{n=1}^{N_c} \hat{x}_n^{(c)}$. ▷ total mean and class mean.
    calculate: $S_W = \sum_{c=1}^{C} \sum_{n=1}^{N_c} (\hat{x}_n^{(c)} - \mu_c)(\hat{x}_n^{(c)} - \mu_c)^\intercal$. ▷ within scatter.
    calculate: $S_B = \sum_{c=1}^{C} N_c(\mu_c - \mu)(\mu_c - \mu)^\intercal$. ▷ between scatter.
    calculate: $\lambda = 1/(1 + \exp^{-a\sigma(S_B)})$. ▷ regularization strength.
    solve U: $S_B U = [\text{tr}(U^\intercal S_B U)/\text{tr}(U^\intercal \tilde{S}_W U)]\tilde{S}_W U$ in Eqn.(4).
    calculate: updated representations $\tilde{x} = U^\intercal \hat{x}$ in Fisher Space.
    calculate: $\delta_c(\hat{x}_n) = \hat{x}_n{}^\intercal U U^\intercal \mu_c - \frac{1}{2}\mu_c{}^\intercal U U^\intercal \mu_c + \log q_c$.
    calculate: $p(y_n|x_n) = \exp^{\delta_{y_n}(\hat{x}_n)} / \sum_{c=1}^{C} \exp^{\delta_c(\hat{x}_n)}$
    if self-challenge **then**
        calculate: $\tilde{x}_n = p_n \hat{x}_n + (1 - p_n)\mu_{c \neq y_n}$ ▷ ConfMix Noise.
        Set: self-challenge = False
        turn to line 7.
    end if
    Calculate: $T_m = \frac{1}{N} \sum_{n=1}^{N} \log p(y_n|x_n)$. ▷ Transferability score of SFDA.
end for



(a) t-SNE of $\hat{x}$ with supervised ResNet-50 (b) t-SNE of $\hat{x}$ with Infomin ResNet-50

Fig. 5. (a & b) show that Infomin ResNet-50 has a larger within scatter of classes than its supervised counterpart on CIFAR-10 dataset.

Algorithm 1 illustrates the whole pipeline of our proposed SFDA. As we can see, SFDA consists of a two-stage Reg-FDA. The Reg-FDA in the second stage challenges the Reg-FDA in the first stage by the proposed ConfMix noise. SFDA has an obvious advantage compared with previous transferability metrics such as LogME, because it explicitly projects static features $\{\hat{x}_n\}_{n=1}^N$ into a Fisher space where features exhibits better linear separability as shown in line 13 in Algorithm 1. Besides, the updated features $\{\hat{x}_n\}_{n=1}^N$ becomes more discriminative in classification difficulty, behaving more like finetuning than other metrics. Moreover, SFDA is fast to obtain, as it requires no gradient optimization and only involves generalized eigenvalue problem.

Note that the largest eigenvalue of $\sigma(S_B)$ in line 11 of Algorithm 1 can be obtained by iteration method [6]. Specifically, we calculate $\sigma(S_B)$ by the following iteration. For $s = 1, 2 \cdots S$,

$$v_s = {S_B}^\mathsf{T} u_{s-1} / \|{S_B}^\mathsf{T} u_{s-1}\|_2 \text{ and } u_s = {S_B}^\mathsf{T} v_s / \|{S_B}^\mathsf{T} v_s\|_2 \tag{11}$$

where we initialize u_0 as a vector of all ones. After S iterations, we have $\sigma(S_B) = {u_S}^\mathsf{T} S_B v_S$. In practice, we find that $S = 3$ is enough for obtaining a precise $\sigma(S_B)$. Note that Eqn.(11) only involves matrix-vector product. $\sigma(S_B)$ can be efficiently acquired.

A.3 Feature visualization by t-SNE

In Sec.4.1, we treat $\lambda \in [0, 1]$ as an adaptive regularization strength, considering the diverse distribution of features $\{\hat{x}_i\}_{i=1}^N$ extracted from different pre-trained models. For example, as shown in Fig.5, supervised ResNet-50 has a larger between scatter of classes than its self-supervised counterpart with Infomin on CIFAR-10 dataset, implying that ResNet-50 with Infomin needs stronger supervision on minimizing within scatter of every class for better classes separation.

Algorithm 2 Top-k model ensembles of SFDA.

Input: projected features in Fisher space $\{(\tilde{x}_n^m\}_{n=1}^{N_{\text{ens}}}, m = 1, 2, \cdots M$ where the superscript m denotes the m-th model and N_{ens} is the number of samples used to select top-k models;
Hyper-parameters: k to the number of selected models, $r = 0.5$ in Eqn.(9).
Output: top-k models ensemble;.
for $n = 1$ **to** N_{ens} **do**
    calculate: models feature ensemble $F_n^{\text{ens}} = [\tilde{x}_n^1, \cdots, \tilde{x}_n^M]$.
    calculate: complementarity score $(T_m^{\text{com}})_n = \|F_n^{\text{ens}}\|_* - \|F_n^{\text{ens}} \odot 1_m\|_*$
    calculate: model ensemble score $(T_m^{\text{ens}})_n = r(T_m^{\text{SFDA}})_n + (1 - r)(T_m^{\text{com}})_n$.
end for
calculate: T_m^{ens} by averaging $(T_m^{\text{ens}})_n$ under all N_{ens} input samples.
ranking: select k models by top-k ranked ensemble $T_m^{\text{ens}}, n = 1, 2, \cdots M$

A.4 Algorithm of SFDA for top-k model ensembles selection

Here we provide the framework of SFDA for top-k model ensembles selection in Algorithm 2. As we can see, the total ensemble score for selecting top-k models is determined by combining SFDA and complementarity scores. The former evaluates the transferability of a single model, and the latter measures the complementarity between models. In experiment, we set $N_{\text{ens}} = 3000$ which is large enough to measure the complementarity score precisely and efficiently (tens second to run Algorithm 2 for each target task).

B More Experimental Results

B.1 Results of ground truth of fine-tuning

Fine-tuning details. The ground-truth of the problem of pre-trained models ranking is to fine-tune all pre-trained models with a hyper-parameters sweep on target datasets. Given the model and the target dataset, two of the most important parameters would be learning rate and weight decay in optimizing the model [4]. Therefore, we careful fine-tune pre-trained models with a grid search of learning rate in $\{1e-1, 1e-2, 1e-3, 1e-4\}$ and weight decay in $\{1e-3, 1e-4, 1e-5, 1e-6, 0\}$. After determining the best hyper-parameters candidate, we fine-tune the pre-trained model on the target dataset with the candidate and then obtain the test accuracy as the ground truth. We use a Tesla V100 with a batch size of 128 to perform finetuning. All input images are resized to 224×224. To avoid random error, we repeat the above fine-tuning procedure three times and take an average to obtain the final fine-tuning accuracy. For reference, we list the fine-tuning accuracy of supervised CNN models in Sec.5.2, self-supervised CNN models in Sec.5.3, and vision transformer models in Sec.B.2 in Table 6, Table 7, and Table 8, respectively. To obtain ensemble finetuning accuracy, we also use the above hyper-parameters sweep. To avoid huge memory consumption, we firstly finetune each pre-trained model on target dataset and then fix the static representation extracted by the fine-tuned pretrained model. After that, we train k classification head for each model, and average the resulting class logits to make the final label prediction.

Table 6. The fine-tuning accuracy of supervised CNN models on 11 target tasks.

	Aircraft	Caltech	Cars	CF-10	CF-100	DTD	Flowers	Food	Pets	SUN	VOC
ResNet-34	84.06	91.15	88.63	96.12	81.94	72.96	95.2	81.99	93.5	61.02	84.6
ResNet-50	84.64	91.98	89.09	96.28	82.8	74.72	96.26	84.45	93.88	63.54	85.8
ResNet-101	85.53	92.38	89.47	97.39	84.88	74.8	96.53	85.58	93.92	63.76	85.68
ResNet-152	86.29	93.1	89.88	97.53	85.66	76.44	96.86	86.28	94.42	64.82	86.32
DenseNet-121	84.66	91.5	89.34	96.45	82.75	74.18	97.02	84.99	93.07	63.26	85.28
DenseNet-169	84.19	92.51	89.02	96.77	84.26	74.72	97.32	85.84	93.62	64.1	85.77
DenseNet-201	85.38	93.14	89.44	97.02	84.88	76.04	97.1	86.71	94.03	64.57	85.67
MNet-A1	66.48	89.34	72.58	92.59	72.04	70.12	95.39	71.35	91.08	56.56	81.06
MobileNetV2	79.68	88.64	86.44	94.74	78.11	71.72	96.2	81.12	91.28	60.29	82.8
Googlenet	80.32	90.85	87.76	95.54	79.84	72.53	95.76	79.3	91.38	59.89	82.58
InceptionV3	80.15	92.75	87.74	96.18	81.49	72.85	95.73	81.76	92.14	59.98	83.84

Table 7. The fine-tuning accuracy of self-supervised CNN models on 11 target tasks.

	Aircraft	Caltech	Cars	CF-10	CF-100	DTD	Flowers	Food	Pets	SUN	VOC
BYOL	82.1	91.9	89.83	96.98	83.86	76.37	96.8	85.44	91.48	63.69	85.13
Deepclusterv2	82.43	91.16	90.16	97.17	84.84	77.31	97.05	87.24	90.89	66.54	85.38
Infomin	83.78	80.86	86.9	96.72	70.89	73.47	95.81	78.82	90.92	57.67	81.41
InsDis	79.7	77.21	80.21	93.08	69.08	66.4	93.63	76.47	84.58	51.62	76.33
MoCov1	81.85	79.68	82.19	94.15	71.23	67.36	94.32	77.21	85.26	53.83	77.94
MoCov2	83.7	82.76	85.55	96.48	71.27	72.56	95.12	77.15	89.06	56.28	78.32
PCLv1	82.16	88.6	87.15	96.42	79.44	73.28	95.62	77.7	88.93	58.36	81.91
PCLv2	83.0	87.52	85.56	96.55	79.84	69.3	95.87	80.29	88.72	58.82	81.85
Sela-v2	85.42	90.53	89.85	96.85	84.36	76.03	96.22	86.37	89.61	65.74	85.52
SimCLRv1	80.54	90.94	89.98	97.09	84.49	73.97	95.33	82.2	88.53	63.46	83.29
SimCLRv2	81.5	88.58	88.82	96.22	78.91	74.71	95.39	82.23	89.18	60.93	83.08
SWAV	83.04	89.49	89.81	96.81	83.78	76.68	97.11	87.22	90.59	66.1	85.06

Hardware for counting wall-clock time. For all wall-clock time counting, we use Intel(R) Xeon(R) Platinum CPU.

B.2 Evaluation on Vision Transformer Models

Models. Vision transformer (ViT) models have attracted much attention recently due to its power in processing multi-modal data. When pre-training ViT models, various model architectures and data augmentation settings result in models with drastically different performance. Hence, how to select a model for further adaptation for an end application is significant in practice. We compare SFDA with other metrics in ranking pre-trained ViT models in terms of transferability. To this end, we collect 10 ViT models including ViT-T [11], ViT-S [11], ViT-B [11], DINO-S [1], MoCov3-S [2] , PVTv2-B2 [7], PVT-T [7], PVT-S [7], PVT-M [7], and Swin-T [5]. The ground truth of models' transferability are obtained by fine-tuning these models on 11 downstream tasks as shown in Table 8.

Performance Comparison. We compare our SFDA with LogME and NLEEP on transferability assessment in terms of rank correlation τ_w. The results are

Table 8. The fine-tuning accuracy of vision transformer models on 11 target tasks.

	Aircraft	Caltech	Cars	CF-10	CF-100	DTD	Flowers	Food	Pets	SUN	VOC
ViT-T	71.26	89.39	82.09	96.52	81.58	71.86	95.5	81.96	91.44	58.4	83.1
ViT-S	73.12	92.7	86.72	97.69	86.62	75.08	96.79	86.26	94.02	64.76	86.62
ViT-B	78.39	93.47	89.26	98.56	89.96	77.66	97.98	88.96	94.61	68.62	87.88
PVTv2-B2	84.14	93.13	90.6	97.96	88.24	77.16	97.89	88.67	93.86	66.44	86.44
PVT-T	69.76	90.04	84.1	94.87	75.26	72.92	95.8	83.78	91.48	61.86	84.6
PVT-S	75.2	93.02	87.61	97.34	86.2	75.77	97.32	86.98	94.13	65.78	86.62
PVT-M	76.7	93.75	87.66	97.93	87.36	77.1	97.36	85.56	94.48	67.22	87.36
Swin-T	81.9	91.9	88.93	97.34	85.97	77.04	97.4	86.67	94.5	65.51	87.54
MoCov3-S	76.04	89.84	82.18	97.92	85.84	71.88	93.89	82.84	90.44	60.6	81.84
DINO-S	72.18	86.76	79.81	97.96	85.66	75.96	95.96	85.69	92.59	64.14	84.8

Table 9. Comparison of different transferability metrics on ViT models in terms of τ_w and the wall-clock time. We see that our proposed SFDA achieves better trade-off between transferability assessment and computation consumption over 11 target tasks.

	Aircraft	Caltech	Cars	CF-10	CF-100	DTD	Flowers	Food	Pets	SUN	VOC
Weighted Kendall's tau τ_w											
LogME	0.299	0.382	0.642	0.741	**0.723**	0.569	0.512	0.580	0.528	0.619	0.519
NLEEP	-0.282	0.027	**0.693**	0.674	0.538	0.123	-0.262	0.105	0.409	0.268	0.109
SFDA	**0.533**	**0.533**	0.632	**0.743**	0.692	**0.570**	**0.515**	**0.592**	**0.787**	**0.707**	**0.809**
Wall-Clock Time (s)											
LogME	5.7	3.6	11.2	13.1	21.9	14.2	3.6	33.1	4.9	186.1	3.6
NLEEP	553.7	716.8	1.1e3	8.0e3	1.2e4	183.7	819.2	3.4e4	256.4	2.7e4	288.3
SFDA	31.3	34.0	59.1	121.7	140.3	20.7	28.7	218.8	45.6	129.8	26.5

reported in Table 9, SFDA still performs consistently well in measuring transferability of self-supervised models. An interesting observation is that LogME outperform NLEEP on evaluation of ViT models, which is not the case on evaluation supervised and self-supervised CNN models. We guess that LogME is adept in deal with sequential feature representation extracted from ViT models as LogME is also designed for regression downstream tasks. Averaging τ_w over 11 target tasks, SFDA (0.647) improves rank correlation τ_w by 196.8% and 16.3% relative to NLEEP (0.218) and LogME (0.556), respectively. Hence, our SFDA can measure the transferability of pre-trained ViT models better.

Wall-clock time comparison. We provide wall-clock time comparison in Table 9. We can see that LogME is efficient enough to calculate transferability score on all target tasks. Moreover, NLEEP performs worse than LogME and our SFDA in terms of both rank correlation τ_w and computation efficiency. In addition, our SFDA is much more efficient in computing transferability score while achieving the best transferability assessment.

Table 10. Comparison of different transferability metrics on top-k model ensembles selection. Results are obtained by selecting top-2 and top-3 models among supervised CNN models and them perform ensemble fine-tuning following [1]. SFDA and SFDA$^{\text{com}}$ select top-k models by top-k ranked SFDA scores and top-k ranked ensemble score through Eqn.(9), respectively. SFDA$^{\text{com}}$ generally performs well over 11 target tasks as it considers complementarity between models.

Top-k	Method	Aircraft	Caltech	Cars	CF-10	CF-100	DTD	Flowers	Food	Pets	SUN	VOC
$k=2$	LogME	87.23	**93.87**	91.38	97.86	**86.98**	77.14	96.87	87.43	**94.62**	65.47	86.46
	NLEEP	85.17	93.45	91.03	97.86	**86.98**	**77.25**	97.04	86.62	94.59	**65.81**	**86.64**
	SFDA	**87.71**	93.40	91.03	97.86	**86.98**	76.95	**97.59**	**87.95**	94.58	**65.81**	86.46
	SFDA$^{\text{com}}$	**87.71**	**93.87**	**91.69**	**97.91**	**86.98**	76.95	97.24	**87.95**	**94.62**	**65.81**	86.46
$k=3$	LogME	87.23	93.87	91.80	97.88	86.96	77.68	97.56	87.83	94.70	66.20	**86.89**
	NLEEP	86.98	**94.12**	91.76	**98.02**	**87.48**	**78.14**	97.65	87.37	94.71	**66.95**	**86.89**
	SFDA	**88.01**	93.95	91.76	**98.02**	**87.48**	77.68	**97.99**	**88.49**	94.82	66.53	**86.89**
	SFDA$^{\text{com}}$	**88.01**	93.87	**91.95**	**98.02**	**87.48**	**78.14**	97.35	**88.49**	**94.92**	66.53	**86.89**

Table 11. SFDA under different measurements of transferability assessment on CIFAR-10 and CIFAR-100 datasets using supervised CNN models.

Data	Method	Rel@1	Rel@3	r	r_w	τ	τ_w	Data	Method	Rel@1	Rel@3	r	r_w	τ	τ_w
CF10	LEEP	**1.0**	**1.0**	0.623	0.753	0.673	0.824	CF100	LEEP	0.991	**1.0**	0.653	0.692	0.624	0.677
	LogME	**1.0**	**1.0**	0.718	0.756	0.782	0.852		LogME	**1.0**	**1.0**	0.508	0.586	0.477	0.692
	NLEEP	**1.0**	**1.0**	0.635	0.774	0.636	0.806		NLEEP	**1.0**	**1.0**	**0.694**	0.762	0.734	0.823
	SFDA	**1.0**	**1.0**	**0.768**	**0.801**	**0.891**	**0.949**		SFDA	**1.0**	**1.0**	0.691	**0.764**	**0.771**	**0.896**

B.3 Results on Top-k Model Ensembles Selection

Performance Comparison. Table 10 shows that SFDA$^{\text{com}}$ leads to higher fine-tuning accuracy on most target tasks. It demonstrates SFDA's superiority to those metrics which do not consider the complementarity between models. For example, when performing top-2 and top-3 ensembles selection, SFDA$^{\text{com}}$ outperforms LogME, NLEEP, and SFDA on 9 and 8 downstream tasks. Therefore, SFDA$^{\text{com}}$ is effective in multiple pre-trained model ensembles selection.

B.4 More ablation study

SFDA under other transferability assessment measures. Other than weighted Kendall's tau, here we also adopt different types of measurement to evaluate our SFDA. The measures include Kendall's tau (τ), Pearson's correlation (r), weighted Pearson's correlation (r_w), top-k relative accuracy denoted as Rel@k that is the ratio between the best fine-tuning accuracy on the downstream task with the top-k ranked models and the best fine-tuning precision with all the models. We test the robustness of transferability metrics to different measurements using supervised CNN models on the CIFAR-10 and CIFAR-100 datasets in Table 11. Our SFDA consistently outperforms previous transferability metrics such as LEEP, LogME and NLEEP under the above measurements, showing the superiority of SFDA.

References

1. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021)
2. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised visual transformers. arXiv e-prints pp. arXiv–2104 (2021)
3. 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. arXiv preprint arXiv:2010.11929 (2020)
4. Li, H., Chaudhari, P., Yang, H., Lam, M., Ravichandran, A., Bhotika, R., Soatto, S.: Rethinking the hyperparameters for fine-tuning. arXiv preprint arXiv:2002.11770 (2020)
5. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
6. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
7. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 568–578 (2021)