Exploring Model Transferability through the Lens of Potential Energy

Xiaotong Li[1,2] Zixuan Hu[1] Yixiao Ge[3] Ying Shan[3] Ling-Yu Duan[1,2*]
[1] School of Computer Science, Peking University, Beijing, China,
[2] Peng Cheng Laboratory, Shenzhen, China, [3] ARC Lab, Tencent PCG, Beijing, China

lixiaotong@stu.pku.edu.cn, {yixiaoge, yingsshan}@tencent.com,
{hzxuan, lingyu}@pku.edu.cn

A. Compared Methods

In this paper, we evaluate the efficacy of our proposed method by applying it to three distinct transferability prediction metrics, *i.e.*, LogME [17], GBC [13], and SFDA [14]. To enhance understanding of their underlying principles and mechanisms, we provide detailed descriptions of these metrics in this section.

LogME [17]. LogME is an evidence-based metric, which uses the marginal evidence to measure the transferability. Unlike the approach in [2], LogME does not directly minimize the Gaussian-based log-likelihood. Instead, it adopts Bayesian averaging to address the overfitting problem:

$$p(y|F) = \int p(w)p(y|F,w)\mathrm{d}w,$$

where $p(w)$ and $p(y|F,w)$ are modeled as two Gaussian distributions specified by two positive parameters. $p(y|F)$ denotes the probability density of the compatibility between features F and labels y, which is based on the marginal evidence of the target task.

SFDA [14]. SFDA is a class-discrimination based metric, which utilizes a Fisher Discriminant Analysis (FDA) approach and propose ConfMix to produce hard-negative samples in a self-challenge manner. The aim of SFDA is to find a transformation U to maximize between scatter of classes and minimize within scatter of each class:

$$U = \arg\max_{U} = \frac{|U^\top S_B U|}{|U^\top (1-\lambda) S_W + \lambda I) U|},$$

where S_B and S_B are the between and within class scatter matrix. The solution can be solved with a close-form solution and then SFDA acquires transformed feature $\{\hat{x}_n = U^n\}_{n=1}^N$. Finally, SFDA adopts Bayes theorem to obtain the score function $\delta_c(\hat{x}_n)$ and use the probability likelihood to measure the transferability score.

$$\delta_c(\hat{x}_n) = \hat{x_n} UU\top\mu_c - \frac{1}{2}\mu_c UU\top\mu_c + \log q_c.$$

GBC [13]. GBC is a class-separation based metric that employs the Gaussian Bhattacharyya Coefficient (GBC) to estimate the pairwise class separability.

$$\begin{aligned}\mathrm{GBC} &= -\sum_{i\neq j}\exp(-\mathrm{BC}(i,j)) \\ \mathrm{BC}(i,j) &= \frac{1}{8}(\mu_{c_i}-\mu_{c_j})^\top \Sigma^\top (\mu_{c_i}-\mu_{c_j}) \\ &\quad + \frac{1}{2}\ln\left(\frac{|\Sigma|}{\sqrt{|\Sigma_{c_i}||\Sigma_{c_j}|}}\right),\end{aligned}$$

where μ and Σ represent the distribution mean and variance of the corresponding class, and coefficient $\mathrm{BC}(i,j)$ denotes the overlaps between classes i and j. The final transferability score is based on the overlaps of all classes by summing up the pairwise negative exponential coefficients.

B. Implementation Details

The implementation details are presented in the section of experiment setup and ablation study. Additionally, we present supplementary studies in this section.

Hyper-parameter k. The hyper-parameter k denotes the elastic coefficient of the repulsive-based elastic force and a higher value of k yields a stronger force, as shown in Section 3.3. Since that the elastic hyper-parameter k is coupled with the radius coefficient λ, we set the default value of the hyper-parameter k to 1.0 and adjust λ in our experiments accordingly. As suggested by [14], we further conduct a grid search on k for optimal performance using values of [0.6, 0.8, 1.0, 1.2, 1.5, 2.0] and the results are presented in Table 1.

*Corresponding Author.

Table 1. The supplementary experiment results of different transferability metrics on various self-supervised learning models under grid-search of hyper-parameter k, showing that our method still has further potential with fine-grained tuning on hyper-parameter.

Self-Supervised	Reference	Aircraft	Caltech101	Cars	Cifar10	Cifar100	Flowers	VOC	Pets	Food	DTD
$\mathcal{N}$LEEP [12]	CVPR'21	-0.029	0.525	0.486	-0.044	0.276	0.534	-0.101	0.792	0574	0.641
PARC [3]	NIPS'21	-0.03	0.196	0.424	0.147	-0.136	0.622	0.618	0.496	0.359	0.447
LogME [17]	ICML'21	0.223	0.051	0.375	0.295	-0.008	0.604	0.158	0.684	0.570	0.627
LogME+Ours	this paper	0.509	0.611	0.624	0.633	0.668	0.728	**0.781**	**0.795**	**0.737**	0.837
SFDA [14]	ECCV'22	0.254	0.523	0.515	0.619	0.548	0.773	0.568	0.586	0.685	0.749
SFDA+Ours	this paper	0.505	**0.661**	**0.666**	**0.741**	**0.744**	**0.798**	0.613	0.592	0.689	**0.907**
GBC [13]	CVPR'22	0.048	-0.18	0.424	0.008	-0.249	0.532	-0.041	0.655	0.268	0.05
GBC+Ours	this paper	**0.549**	0.340	0.629	0.149	0.431	0.779	0.552	0.758	0.672	0.611

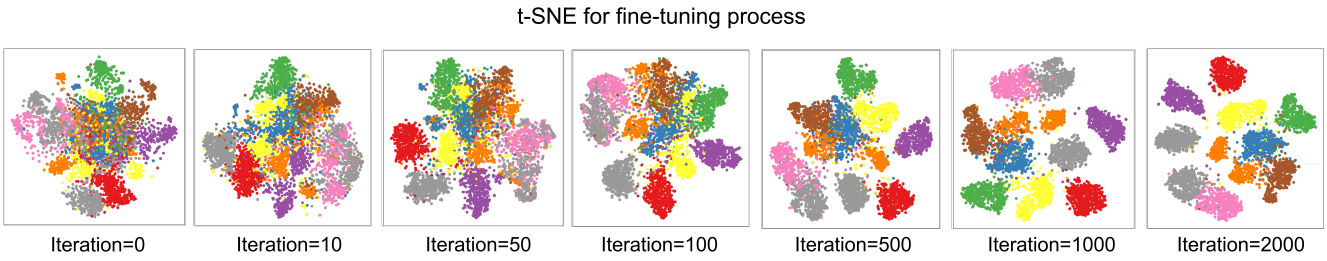

Figure 1. The t-SNE visualization of dynamic feature representation achieved through fine-tuning.

Feature Pre-processing. In our implementation, we adopt a pre-processing step to analyze the motion in the embedding space. Specifically, downstream features are normalized with ImageNet feature mean and standard deviation, based on a subset of 50,000 images. To evaluate the impact of normalization on the modeling process, we present our findings in Fig 2. Through the normalization, the feature values in each dimension are largely normalized in a certain region (*e.g.,* $[-3\sigma, 3\sigma]$ due to the property of Gaussian distribution), creating a suitable condition for physical modeling. We discovered that the normalization can prevent the occurrence of highly imbalanced dimensions caused by the divergence in numerical value and stabilize the physical modeling process.

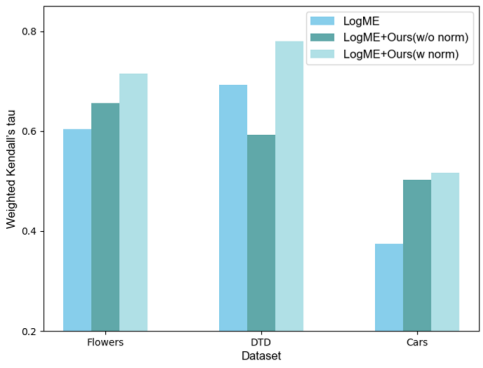

Figure 2. The influence of feature pre-processing.

Maximum Phase Position. We calculate the phase position s of each cluster using the motion equation. In our implementation, we set a maximum phase position constraint, i.e., $\min(s, x_e)$. This constraint ensures that the force decreases accordingly as the movement s surpasses the overlap x_e between two clusters. By adding this boundary condition to the motion equation, we can enhances the algorithm's robustness and avoid extreme situations.

C. Comparing to Fine-tuning

In this paper, we have shown the effectiveness of our method without the need for fine-tuning, Additionally, we highlight the advantages of our method over fine-tuning in this section. The t-SNE visualization in Fig. 1 reveals that during the initial stage of fine-tuning, the clusters are not well separated due to the random initialized classifier layer and further adaptation to downstream tasks is required. For comparison, the t-SNE visualization of our approach is shown in Section 6.1.

We display the ranking performance achieved through fine-tuning in Fig. 3, which reveals a performance pattern of initial decline followed by improvement. This suggests that fine-tuning requires multiple iterations to adapt to new tasks for learning the classifier layer. In contrast, our proposed physics-inspired method can simulate the dynamic feature representation without the need for this adaptation process.

Furthermore, fine-tuning involves a grid search strategy to select the best hyper-parameters, and fine-tuning the entire model on the downstream dataset. This process requires

Table 2. The ground truth results of the 12 self-supervised pre-trained models on 10 downstream tasks.

Self-Supervised	Aircraft	Caltech101	Cars	Cifar10	Cifar100	Flowers	VOC	Pets	Food	DTD
BYOL [9]	82.1	91.9	89.83	96.98	83.86	96.8	85.13	91.48	85.44	76.37
Deepclusterv2 [4]	82.43	91.16	90.16	97.17	84.84	97.05	85.38	90.89	87.24	77.31
Infomin [15]	83.78	80.86	86.9	96.72	70.89	95.81	81.41	90.92	78.82	73.74
InsDis [16]	79.7	77.21	80.21	93.08	69.08	93.63	76.33	84.58	76.47	66.4
MoCov1 [10]	81.85	79.68	82.19	84.15	71.23	94.32	77.94	85.26	77.21	67.36
MoCov2 [8]	83.7	82.76	85.55	96.48	71.27	95.12	78.32	89.06	77.15	72.56
PCLv1 [11]	82.16	88.6	87.15	86.42	79.44	95.62	91.91	88.93	77.7	73.28
PCLv2 [11]	83.0	87.52	85.56	96.55	79.84	95.87	81.85	88.72	80.29	69.3
Sela-v2 [1]	85.42	90.53	89.85	96.85	84.36	96.22	85.52	89.61	86.37	76.03
SimCLRv1 [6]	80.54	90.94	89.98	97.09	84.49	95.33	83.29	88.53	82.2	73.97
SimCLRv2 [7]	81.5	88.58	88.82	96.22	78.91	95.39	83.08	89.18	82.23	94.71
Swav [5]	83.04	89.49	89.81	96.81	83.78	97.11	85.06	90.59	87.22	76.68

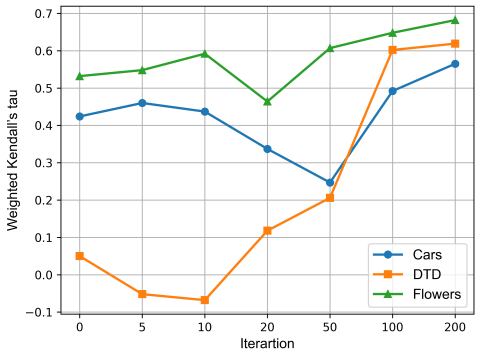

Figure 3. The ranking performance of different fine-tuning iterations.

testing 30 hyper-parameter setups, with each training process consisting of 5000 iterations taking 16 minutes, making it more time-consuming compared to our physics-driven approach.

D. Ground Truth Results

We obtained the ground truth results by fine-tuning the models using a grid-search strategy, following the the implementation of [14, 17]. More information on this process can be found in Section 4. In Table 2, we present the ground truth results of the 12 self-supervised learning models and 10 downstream tasks.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 3

[2] Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *ICIP*, pages 2309–2313, 2019. 1

[3] Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. Scalable diverse model selection for accessible transfer learning. *NeurIPS*, pages 19301–19312, 2021. 2

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 3

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33:9912–9924, 2020. 3

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 3

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *NeurIPS*, pages 22243–22255, 2020. 3

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[9] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, pages 21271–21284, 2020. 3

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 3

[11] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 3

[12] Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. Ranking neural checkpoints. In *CVPR*, pages 2663–2673, 2021. 2

[13] Michal Pándy, Andrea Agostinelli, Jasper Uijlings, Vittorio Ferrari, and Thomas Mensink. Transferability estimation using bhattacharyya class separability. In *CVPR*, pages 9172–9182, 2022. 1, 2

[14] Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. Not all models are equal: Predicting model transferability in a self-challenging fisher space. In *ECCV*, pages 286–302, 2022. 1, 2, 3

[15] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *NeurIPS*, 33:6827–6839, 2020. 3

[16] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 3

[17] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *ICML*, pages 12133–12143, 2021. 1, 2, 3