Tuning Stable Rank Shrinkage: Aiming at the Overlooked Structural Risk in Fine-tuning

Supplementary Material

7. Theorectical proof of Eq. (7)

Directly computing the stable rank of the weight matrix of an entire NN is difficult. Therefore, we propose to use noise sensitivity as an approximation of the stable rank for estimating the model complexity. As shown by Arora *et al.* [3], there is a strong correlation between noise sensitivity and stable rank of a model. Concretely, low noise sensitivity corresponds to a weight matrix having some large singular values, i.e. having a lower stable rank. Given a Gaussian noise η with noise intensity ϕ, the noise sensitivity of model M with weight w is defined as:

$$\psi(M,x) = \mathop{\mathbb{E}}_{\eta\in\mathcal{N}}\left[\frac{\|M(x+\eta\|x\|,w)-M(x,w)\|^2}{\|M(x,w)\|^2}\right]. \tag{9}$$

Given $\eta \in \mathcal{N}(0,\phi^2)$, we have the following guarantee: $\mathrm{srank}(w) \le \phi^2\psi(M,x)_w$. Therefore, noise sensitivity is the upper bound of stable rank [3], i.e. the model complexity can be estimated by noise sensitivity. The detailed proof is given below.

For the liner layer, we have:

$$y = Wx, \tag{10}$$

and

$$y' = W(x+\eta\|x\|). \tag{11}$$

Because we assume that η is sub-Gaussian with ϕ, we have:

$$E\left[\eta\eta^\top\right] = \phi^2 I. \tag{12}$$

The noise sensitivity is:

$$\begin{aligned}
\psi(Wx,x) &= E\left[\frac{y'-y}{y}\right] \\
&= E_\eta\left[\frac{\|W(x+\eta\|x\|)-Wx\|^2}{\|Wx\|^2}\right] \\
&= E_\eta\left[\frac{\|x\|^2\|W\eta\|^2}{\|WX\|^2}\right] \\
&= E_\eta\left[\frac{\|x\|^2\,\mathrm{tr}\left(W\eta\eta^\top W\right)}{\|Wx\|^2}\right] \\
&= \phi^2\frac{\|x\|^2}{\|Wx\|^2}\,\mathrm{tr}\left(WW^\top\right) \\
&\ge \phi^2\frac{\|x\|^2\|W\|_F^2}{\|x\|^2\|W\|_2^2} \\
&= \phi^2\frac{\|W\|_F^2}{\|W\|_2^2} \\
&= \phi^2\,\mathrm{srank}(W).
\end{aligned} \tag{13}$$

Table 6. Classification results on multiple datasets.

Method	Caltech101	CIFAR-100	DTD	EuroSAT	Flowers102
Baseline	92.9	84.1	74.4	98.8	96.5
BSS	93.0	84.2	74.3	98.8	96.6
BSS+TSRS	**93.1**	**85.0**	**74.6**	**98.8**	**96.8**

Method	PACM	Resisc45	SVHN	Food101	Pets	Average
Baseline	83.8	96.1	96.7	85.1	93.2	90.2
BSS	83.9	96.1	96.8	85.0	93.3	90.2
BSS+TSRS	**84.6**	**96.2**	**96.8**	**85.8**	**93.7**	**90.5**

Assume the model M employs a piecewise-linear activation function, like ReLU, Then M, as a combination of linear and piecewise-linear functions, is also a piecewise-linear mapping [43]. So we have:

$$\mathrm{srank}(w) \le \mathcal{O}\left(\psi(M,x)_w\right). \tag{14}$$

8. More validation datasets on classification

We summarize the results on other **10** datasets for validation in Tab. 6. These encompass natural image datasets (Caltech101, CIFAR-100, DTD, Flowers102, Pets, Food101, SVHN), remote sensing datasets (Resisc45, EuroSAT), and the medical dataset PACM, most of which are part of VTAB-1k [61]. All experiments in Tab. 6 were conducted on ImageNet-pre-trained ResNet-50 using 100% tuning data based on the TLlib open-source library.

9. More results on Detection

We have added an additional validation dataset for the detection task: DeepLesion dataset [58] (Tab. 7). The experimental settings for these results align with the segmentation and detection settings outlined in our main text.

Table 7. Detection results on the DeepLesion dataset.

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
$DETR_{base}$	46.42	48.16	19.69	41.58	48.11	50.86
$DETR_{base}$+TSRS	**47.34**	**49.16**	**20.18**	**41.95**	**48.82**	**54.30**

10. Detailed formulation of existing fine-tuning methods

M, w_t denote the model and its target weights. $D_s = \{(x_s^i, y_s^i)\}_{i=1}^{m_s}$ and $D_t = \{(x_t^i, y_t^i)\}_{i=1}^{m_t}$ denote the source

domain data and the target domain data. Denote the **empirical risk** and the penalty loss on model complexity by R_{emp}, L_{srm}, the **structural risk** is formularized as:

$$R_{srm} = R_{emp} + L_{srm}. \tag{15}$$

In the fine-tuning scenario, the empirical risk can be divided into two parts: one is the empirical risk of newly learned knowledge in the target domain D_t, denoted as L_{emp}, and the other is the empirical risk of inherited knowledge from the source domain, denoted as L_{temp}, which is formularized as:

$$R_{emp} = L_{emp}\left(x_t, y_t, w_t\right) + L_{temp}\left(x_t, y_t, w_t, w_s\right), \tag{16}$$

$$L_{emp} = \frac{1}{m_t}\sum_{i=1}^{m_t} L\left(M\left(x_t^i, w_t\right), y_t^i\right), \tag{17}$$

where $L(\cdot)$ denotes the loss function and cross-entropy is a typical choice. Existing fine-tuning methods aim to reduce R_{emp} and can be unified under the concept of knowledge re-weighting. We take the classification task as an example, and the model consists of a feature extractor F and a classifier G, with weights $\hat{w}$ and $\bar{w}$, respectively. And λ denotes the trade-off weighting parameter.

DELTA. DELTA [33] aligns the outer feature maps of the fine-tuned model and the pre-trained model, ensuring the useful semantic information retainment. The optimization objective of DELTA is formularized as:

$$\begin{aligned} &L_{temp}\left(x_t, y_t, \hat{w}_t, \hat{w}_s\right) \\ &= \lambda \sum_{i=1}^{m_t}\sum_{j=1}^{N} C_j \left\| F_j\left(x_t^i, \hat{w}_t\right) - F_j\left(x_t^i, \hat{w}_s\right)\right\|_2^2, \end{aligned} \tag{18}$$

where C_j refers to the weight assigned to the j^{th} channel, which is computed by the behavioral difference between the two feature maps, and is the core of knowledge re-weighting in DELTA:

$$C_j = \mathrm{softmax}(L(M(x_t^i, \hat{w}_{s\setminus j}), y_t^i) - L(M(x_t^i, \hat{w}_s), y_t^i)). \tag{19}$$

Co-tuning. Co-tuning [60] first learns a relationship $p\left(y_s|y_t\right)$ between the source and target categories and then translates the target domain labels into probability labels on the source label domain, by which it preserves the knowledge in the pre-trained task layer $G(\cdot, \bar{w}_s)$. The optimization objective of Co-tuning is formularized as:

$$L_{\text{temp}} = \lambda \sum_{i=1}^{m_t} L\left(G\left(F\left(x_t^i, \hat{w}_t\right), \bar{w}_s\right), p\left(y_s|y_t = y_t^i\right)\right). \tag{20}$$

Through the probability label $p\left(y_s|y_t = y_t^i\right)$, Co-tuning retains the conducive knowledge in $\bar{w}_s$.

UOTS. UOTS [37] selects source data by similarity to the target data and reuses the selected source data to retain knowledge related to the target domain in fine-tuning. Its optimization objective is formularized as:

$$L_{temp} = \lambda \sum_{i=1}^{m_s} L\left(M\left(x_s^i, w_t\right), y_s^i\right). \tag{21}$$

The approach of adding selected source data to the fine-tuning process is essentially a re-weighting of knowledge.

BSS. BSS [8] penalizes small singular values of the learned representation for suppressing mischievous features, which corresponds to improving fine-tuning by suppressing the detrimental knowledge. The optimization objective is formularized as:

$$L_{temp} = \lambda \sum_{i=1}^{k} \sigma_{-i}^2, \tag{22}$$

where σ refers to the singular value of the learned representation and k is the number of singular values to be penalized.

11. Detailed implementation and hyper-parameters

Our experiments were conducted using PyTorch on four Nvidia RTX 3090 GPUs. The hyper-parameters for fine-tuning were selected based on the validation sets, and we report the hyper-parameter settings that achieved the best accuracy in Tab. 8 and Tab. 9. The data splitting for the training, validation, and testing sets follows the setup proposed by You *et al.* [60].

Tab. 8 presents the selected learning rate and the trade-off weighting parameter λ for L_{temp} in the original methods. The learning rate was searched over the values [0.03, 0.01, 0.003, 0.001, 0.0005] using a grid search. During fine-tuning, the final task layer was trained from scratch, with its learning rate set to be 10 times those of the fine-tuned layers, following the approach of Yosinski *et al.* [59]. All models were optimized by SGD with 0.9 momentum and 0.0005 weight decay for 20 epochs. The learning rates were reduced by a factor of 1/10 in the 12^{th} epoch. The selection criterion for λ ensures that L_{temp} and L_{emp} are of the same magnitude when the model converges. Other hyper-parameters that were not specifically mentioned were kept the same as those in the original paper.

When applying TSRS, we utilized the parameters from Table Tab. 8 and performed additional grid searches for the unique hyper-parameters of TSRS, namely the starting block l, noise intensity ϕ, and weighting hyper-parameter α. The starting block l was searched over the block numbers, ϕ was searched over the values [0.1, 0.03, 0.01], and α was searched over [1, 0.1]. For TSRS, the hyper-parameter vector is denoted as [l, ϕ, α], and the values are recorded in Table Tab. 9.

Table 8. Basic hyper-parameter settings ([learning rate, λ]).

Dataset	Method	ResNet50	ViT-B
CUB	**Baseline**	[0.01, NA]	[0.01, NA]
	BSS	[0.01, 0.001]	[0.01, 0.001]
	Co-Tuning	[0.01, 2.3]	[0.01, 1.0]
	DELTA	[0.01, 0.01]	[0.01, 0.01]
	UOTS	[0.003,1]	[0.0005,1]
Cars	**Baseline**	[0.01, NA]	[0.001, NA]
	BSS	[0.01, 0.001]	[0.001, 0.001]
	Co-Tuning	[0.01, 2.3]	[0.001, 1.0]
	DELTA	[0.01, 0.01]	[0.001, 0.01]
	UOTS	[0.03,0.3]	[0.001,1]
Aircraft	**Baseline**	[0.01, NA]	[0.001, NA]
	BSS	[0.01, 0.001]	[0.001, 0.001]
	Co-Tuning	[0.01, 2.3]	[0.001, 1.0]
	DELTA	[0.01, 0.01]	[0.001, 0.01]
	UOTS	[0.01,0.3]	[0.001,0.5]

Table 9. Hyper-parameter settings of TSRS ([l, ϕ, α]).

Dataset	Method	ResNet50	ViT-B
CUB	**Baseline**	[3, 0.2, 1]	[10, 0.1, 1]
	BSS	[3, 0.2, 1]	[10, 0.1, 1]
	Co-Tuning	[3, 0.2, 1]	[5, 0.01, 0.1]
	DELTA	[3, 0.2, 1]	[5, 0.01, 1]
	UOTS	[3, 0.2, 1]	[5, 0.01, 1]
Cars	**Baseline**	[3, 0.2, 1]	[5, 0.01, 0.1]
	BSS	[3, 0.2, 1]	[10, 0.01, 0.1]
	Co-Tuning	[3, 0.2, 1]	[10, 0.1, 0.1]
	DELTA	[3, 0.2, 1]	[5, 0.01, 0.1]
	UOTS	[3, 0.2, 1]	[5, 0.01, 1]
Aircraft	**Baseline**	[3, 0.2, 1]	[5, 0.1, 1]
	BSS	[3, 0.2, 1]	[10, 0.1, 1]
	Co-Tuning	[3, 0.2, 1]	[10, 0.1, 1]
	DELTA	[3, 0.2, 1]	[5, 0.1, 1]
	UOTS	[3, 0.2, 1]	[5, 0.03, 0.1]

The hyper-parameters were searched on a separate validation set. Once the hyper-parameters were determined, we combined the training and validation sets to form the final training set. The test accuracy reported in Tab. 1 of the main submission is obtained using these trained models.

12. Detailed explanation of performance change with respect to l

Fig. 4c in the main submission presents the experimental results regarding the variation of l, which represents the starting block of the applied constraint. The performance of the model initially exhibited improvement and then reached a stable state as the constraint starting block transitioned from shallow to deep layers.

This observed behavior can be attributed to the following reason. The introduction of noise to the input inherently induces input divergence. In the blocks where the L_{TSRS} constraint is applied, all outputs corresponding to the diverged inputs are constrained to a common point. This constraint effectively leads to compression and clustering of the data space. The process is visually depicted in Fig. 8. Therefore, at the shallow blocks, due to insufficient feature extraction, the distance between intra-class features may be larger than that of inter-class features. This causes the L_{TSRS} constraint added in the shallow blocks to force the features of different classes' samples to cluster together, ultimately impairing the model's performance. Supporting evidence for this assumption is provided in Fig. 5 of the main submission, which presents a visual case.

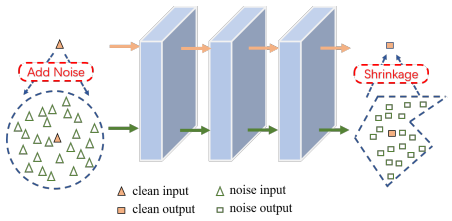

Figure 8. Illustration of clustering effect. The behavior of Tuning Stable Rank Shrinkage (TSRS) can be described as the process of intentionally diverging from the clean input and subsequently constraining the noise output to align with the clean output in the feature space. This approach effectively clusters the data space around the clean input, resulting in the desired effect of shrinkage.

References

[1] Ethem Alpaydin. *Machine learning*. Mit Press, 2021. 2

[2] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. *Medical physics*, 38(2):915–931, 2011. 7

[3] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pages 254–263. PMLR, 2018. 3, 4, 1

[4] José-Ramón Cano. Analysis of data complexity measures for classification. *Expert systems with applications*, 40(12): 4820–4831, 2013. 1

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 7

[6] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021. 7

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2

[8] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 3, 5, 6

[9] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021. 8

[10] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019. 7

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[12] Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. *Advances in neural information processing systems*, 31, 2018. 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[14] RCNN Faster. Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 9199(10.5555):2969239–2969250, 2015. 2

[15] Patrick Glandorf, Timo Kaiser, and Bodo Rosenhahn. Hypersparse neural networks: Shifting exploration to exploitation through adaptive regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1234–1243, 2023. 6

[16] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[18] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019. 1, 2

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1, 2

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[21] Xia Hu, Weiqing Liu, Jiang Bian, and Jian Pei. Measuring model complexity of neural networks with curve activation functions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1521–1531, 2020. 1

[22] Xia Hu, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. Model complexity of deep learning: A survey. *Knowledge and Information Systems*, 63:2585–2619, 2021. 3

[23] Hang Hua, Xingjian Li, Dejing Dou, Chengzhong Xu, and Jiebo Luo. Noise stability regularization for improving bert fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3229–3241, 2021. 3

[24] Junguang Jiang, Yang Shu, Jianmin Wang, and Mingsheng Long. Transferability in deep learning: A survey, 2022. 5

[25] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 3

[26] 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. *arXiv preprint arXiv:1609.04836*, 2016. 8

[27] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan,

John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526, 2017. 1, 2, 3

[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 5, 6

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7

[30] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 8

[31] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018. 8

[32] Ling Li. *Data complexity in machine learning and novel classification algorithms*. California Institute of Technology, 2006. 1

[33] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, Zeyu Chen, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. *arXiv preprint arXiv:1901.09229*, 2019. 1, 2, 6

[34] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd international conference on artificial intelligence and statistics*, pages 888–896. PMLR, 2019. 3

[35] Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403, 2021. 2

[36] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pages 2736–2744, 2017. 2

[37] Ziquan Liu, Yi Xu, Yuanhong Xu, Qi Qian, Hao Li, Xiangyang Ji, Antoni Chan, and Rong Jin. Improved fine-tuning by better leveraging pre-training data. *Advances in Neural Information Processing Systems*, 35:32568–32581, 2022. 1, 2, 6

[38] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5, 6

[39] Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774, 2018. 2

[40] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017. 3

[41] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017. 4

[42] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004. 6

[43] Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018. 3, 1

[44] Namuk Park and Songkuk Kim. Blurs behave like ensembles: Spatial smoothings to improve accuracy, uncertainty, and robustness. In *International Conference on Machine Learning*, pages 17390–17419. PMLR, 2022. 8

[45] Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022. 8

[46] Salah Rifai, Xavier Glorot, Yoshua Bengio, and Pascal Vincent. Adding noise to the input of a model trained with a regularized objective. *arXiv preprint arXiv:1104.3250*, 2011. 7

[47] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21–es, 2007. 3

[48] Amartya Sanyal, Philip HS Torr, and Puneet K Dokania. Stable rank normalization for improved generalization in neural networks and gans. *arXiv preprint arXiv:1906.04659*, 2019. 3, 6

[49] Zhouxing Shi, Yihan Wang, Huan Zhang, J Zico Kolter, and Cho-Jui Hsieh. Efficiently computing local lipschitz constants of neural networks via bound propagation. *Advances in Neural Information Processing Systems*, 35:2350–2364, 2022. 8

[50] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010. 1, 2, 3

[51] Vladimir Vapnik. Principles of risk minimization for learning theory. *Advances in neural information processing systems*, 4, 1991. 1, 2, 3

[52] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999. 2

[53] LN Vasershtein. Stable rank of rings and dimensionality of topological spaces. *Functional Analysis and its Applications*, 5(2):102–110, 1971. 3

[54] Diego Vidaurre, Concha Bielza, and Pedro Larranaga. A survey of l1 regression. *International Statistical Review*, 81 (3):361–387, 2013. 6

[55] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 4, 5, 6

[56] Yi Wang, Zhen-Peng Bian, Junhui Hou, and Lap-Pui Chau. Convolutional neural networks with dynamic regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):2299–2304, 2020. 6

[57] LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pages 2825–2834. PMLR, 2018. 1, 2

[58] Ke Yan, Xiaosong Wang, Le Lu, and Ronald M Summers. Deeplesion: automated mining of large-scale lesion annotations and universal lesion detection with deep learning. *Journal of medical imaging*, 5(3):036501–036501, 2018. 1

[59] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014. 5, 2

[60] Kaichao You, Zhi Kou, Mingsheng Long, and Jianmin Wang. Co-tuning for transfer learning. *Advances in Neural Information Processing Systems*, 33:17236–17246, 2020. 1, 2, 5, 6

[61] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 1

[62] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[63] Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 2022. 1

[64] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020. 1, 2