
A DETAILED IMPLEMENTATION

A.1 DATASET

ImageNet-C ImageNet-C is constructed from the validation set of the original ImageNet, which includes 50,000 samples across 1,000 classes (each class containing 50 samples). Based on the ImageNet validation set, we apply different corruption techniques to create 15 different degradation versions, categorized into four main groups: noise, blur, weather, and digital. For each corruption type, there are five levels of severity, ranging from 1 to 5. Severity level 1 represents the smallest change in the intensity of corruption, while level 5 represents the most significant changes

Imbalance data simulation Our imbalanced data is based on SAR (Niu et al., 2023), and the simulation process can be described as follows: During the adaptation, assume we have a total of T time-steps, where T equals the number of classes C . We set the probability vector $Q_t(y) = [q_1, q_2, \dots, q_C]$, where $q_C = q_{max}$ if $c = t$ and $q_c = q_{min} \triangleq (1 - q_{max})/(C - 1)$ if $c \neq t$. Here, q_{max}/q_{min} represents the imbalance ratio. After that, for each time step $t \in \{1, 2, \dots, T = C\}$, we sample M images from the test set according to $Q_t(y)$. Then, based on the ImageNet-C (Gaussian Noise), we generate a new testing set that has online imbalanced label distribution shifts with a total of $100(M) \times 1000(T)$ images. To achieve this, we need to pre-shuffle the class orders in ImageNet-C because the classes will appear randomly in practice.

CIFAR-10-C and CIFAR-100-C Similar to ImageNet-C, CIFAR-10-C, and CIFAR-100-C are also created from the CIFAR validation set (10,000 samples). The corruption type and corruption level are the same as the ImageNet-C version, which means we also have 15 different corruption types, and each of them also includes 5 levels of severity.

VisDA-21 In this setting, we work on the validation set from the Visual Domain Adaptation Challenge in 2021 (Bashkirova et al., 2022), which contains a subset of images from four different datasets: ImageNet-R, ImageNet-C, ImageNet-O, and ObjectNet. Instead of using all data samples, we use images from ImageNet- $\{R, C, O\}$, which includes a total of 18,338 images.

A.2 MODEL

Architecture For a fair comparison, we utilize existing architectures, which vary across datasets. For ImageNet-C, following SAR, our main architecture is ViTbase-LN from **timm** (Wightman, 2019). The model architectures for CIFAR-10-C and CIFAR-100-C are a 26-layer residual network (He et al., 2016) and WideResNet-18-2 (Diffenderfer et al., 2021), respectively.

Optimizer We use stochastic gradient descent (SGD) as our default optimizer, with the learning rate varying depending on the dataset. Specifically, we set the learning rates for the CIFAR and ImageNet datasets to 0.01 and 0.1, respectively. The momentum is consistent across these datasets, set at 0.9. Additionally, for the sharpness-aware loss, we set the learning rate for SAM to 0.1 in all settings.

A.3 HYPER PARAMETER

s : Number of skipping classes in k -NL In our method, the skipping module balances the risk of noise and useful information in the negative loss. Skipping more negative samples can help the target model reduce noisy information more effectively, but it also filters out valuable negative samples, potentially slowing down the convergence of the training model. We set s equal to 5 in all settings (this value is selected based on cross-validation).

k : Number of selected negative classes in k -NL Generally, utilizing a large number of negative classes can provide more information for the negative loss. However, as shown in (Feng et al., 2020), increasing the number of k also makes the network harder to train. Therefore, the number of negative samples is selected based on cross-validation and is set to 5 in all settings.

Model	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
TENT†	19.8	22.1	52.1	56.9	56.5	61.2	58.0	7.0	6.6	72.2	77.4	67.2	63.1	71.9	69.1	50.7
SAR†	45.8	43.2	45.7	53.5	50.3	57.6	52.6	59.0	54.2	68.8	76.3	65.7	57.8	69.0	66.1	57.7
SAR* + $\mathcal{L}^{sparse-CL}$	52.6	52.5	53.8	56.4	56.3	61.8	59.9	65.4	64.0	72.2	76.7	67.0	66.2	71.7	69.0	63.0
SAR* + \mathcal{L}^{final}	54.9	55.4	56.2	58.2	58.7	63.9	62.8	67.9	65.9	73.4	77.3	68.0	68.6	72.9	70.2	65.0

Table 1: This is the biggest challenge we use to verify our model’s learning ability. In this setting, we work with imbalanced data and small batch sizes, which are common in real-world applications. The results in the table highlight the well-adapted capability of our method, showing that it boosts SAR performance by 5.3% and 7.3% under sparse-CL with and without negative learning, respectively.

Model	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
TENT†	45.0	43.4	45.5	52.4	48.2	55.6	51.3	26.7	24.0	66.7	75.2	64.9	54.0	67.1	64.7	52.3
$\mathcal{L}^{sparse-CL}$	52.0	52.2	53.2	55.5	56.0	61.0	59.5	64.9	63.2	71.2	76.2	66.3	66.0	70.8	68.4	62.4
\mathcal{L}^{final}	54.3	54.7	55.6	57.9	58.2	63.1	61.5	66.2	10.0	72.8	77.3	67.7	67.9	72.6	69.9	60.6

Table 2: Performance of our loss function versus entropy minimization when adapting alone (without high entropy sample filtering, sharpness-aware loss, or model recovery). Generally, adaptation using our loss helps the model transfer better to the target domain, improving target accuracy. Moreover, it helps overcome the failed cases of self-entropy loss (Snow and Frost). Additionally, applying negative learning boosts sparse-CL to achieve better performance on almost all corruption types. However, under some particularly challenging settings (like Frost in this situation), the effect of noise in the negative loss can reduce model performance.

η : Learning rate Selecting the right value for the learning rate is crucial to the success of our method. As mentioned earlier, the learning rate needs to be large enough to help the model converge to a good solution before the negative effects of noisy labels (confirmation bias and memorization) appear. In our main experiments, we empirically found that our learning rate can be up to 100 times larger than the base learning rate in SAR under SAM loss (base learning rate equal to 10^{-2}), or it could be 5000 times larger under the SGD loss when adapting to the TENT setting (base learning rate equal to 10^{-3}).

B ANALYSIS

B.1 THE BENEFIT OF SPARSE UPDATING

Under the prototype learning setting (where we view the classifier $h(\cdot)$ as a list of source prototypes), the proposed sparse-CL utilizes information from the most similar prototype (highest probability class), assigning zero weight to all remaining classes. Therefore, during backpropagation, the gradient only flows through the highest prototype, resulting in sparse updating. Previous works (Iofinova et al., 2022; Andriushchenko et al., 2023; Hoefer et al., 2021) have shown that sparse updating helps the model learn more stably. Moreover, we hypothesize that under a large learning rate, the sparse loss will help the model converge more easily because it only needs to focus on one specific class. Additionally, sparse updating supports faster training (fewer parameters need to be updated), making it beneficial for online learning.

B.2 ON THE CONNECTION BETWEEN OURS AND TRIPLE LOSS

Basically, the classifier $h_t(\cdot)$ is a template matching module that measures the inner product between input features and source prototypes (learned during source domain training). In sparse-CL loss, the model selects the most similar prototype as the positive sample and pulls the feature f_x closer to this (positive) prototype. Conversely, the k -NL loss selects the k -hard negative prototypes to push f_x away. The behavior of our loss during learning is similar to the triplet loss (Schroff et al., 2015; Sohn, 2016; Chen & He, 2021), where f_x is viewed as the anchor, and negative or positive samples are selected for f_x in the prototype set using the inner product operation. This connection partially explains why our loss can help the model converge to better solutions during adaptation. Previous work (Koch et al., 2015) has shown that triplet loss is an efficient loss in few-shot learning, which can be understood as a branch of domain adaptation.

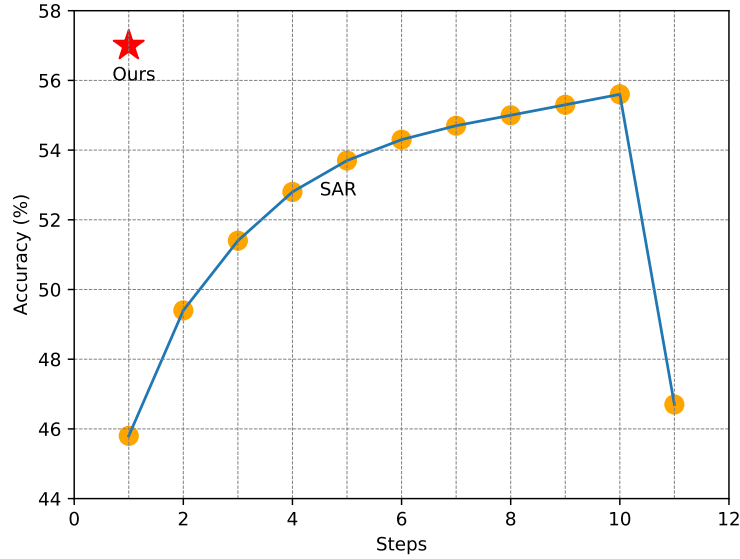


Figure 1: We verify how SAR gradually improves with increasing learning steps on shot noise corruption (severity level 5). The results indicate that SAR stability improves as we increase the number of steps. The enhancement is consistent until the step number reaches 10 (achieving 55.6% accuracy, with our model achieving 57.2% accuracy after 1 step). However, performance deteriorates when the step number reaches 11 (dropping to 46.7% accuracy).

Model	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
TENT†	72.4	74.3	64.1	87.3	65.2	85.9	87.9	82.9	82.8	85.0	91.6	87.7	76.4	80.7	73.0	79.8
SAR†	75.3	75.9	66.8	87.3	64.9	84.9	88.4	84.8	82.6	87.5	91.5	88.2	77.1	83.4	77.1	81.0
\mathcal{L}_{final}	78.1	79.5	70.6	88.0	70.5	86.7	89.2	85.2	84.7	87.9	91.8	89.5	78.0	84.5	78.5	82.8
SAR* + \mathcal{L}_{final}	78.5	78.8	69.0	87.6	69.9	86.0	88.8	84.7	84.5	87.6	91.7	89.8	77.8	83.6	78.5	82.5

Table 3: Results on CIFAR-10-C using our method show the following: First, compared to TENT, SAR performs worse in this setting. Second, our method helps the model improve by over 2.7% compared to TENT and by 1.5% under the SAR setting.

Model	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
TENT†	61.8	63.1	59.1	74.6	59.5	72.8	73.3	67.7	67.7	63.3	74.7	72.6	64.7	68.7	62.1	67.0
SAR†	64.0	65.6	66.0	75.3	62.3	73.8	74.5	69.7	69.5	67.7	75.8	75.5	66.8	71.5	65.3	69.6
\mathcal{L}_{final}	64.3	65.6	65.7	74.6	62.2	73.5	74.1	69.9	69.6	67.7	75.2	74.2	66.6	71.4	65.1	69.3
SAR* + \mathcal{L}_{final}	64.6	66.0	66.8	75.2	62.3	74.3	74.6	70.0	69.9	67.5	75.6	75.0	66.8	72.0	65.2	69.7

Table 4: Under CIFAR-100-C, adapting using Ours does not gain a clear improvement when working with SAR. However, training it alone still helps the model improve up to 2.3% (compared to entropy minimization).

Model	Accuracy
TENT† (Wang et al., 2020)	46.4
\mathcal{L}_{final}	<u>47.9</u>
SAR† (Niu et al., 2023)	46.4
SAR* + \mathcal{L}_{final}	54.6

Table 5: Result of our method on VisDA-21 when learns on normal settings.

Model	Accuracy
TENT [†] (Wang et al., 2020)	47.5
\mathcal{L}_{final}	49.9
SAR [†] (Niu et al., 2023)	47.4
SAR* + \mathcal{L}_{final}	55.4

Table 6: Result of our method on VisDA-21 when learns on small batch size settings.

C ABLATION STUDY

C.1 LEARNING UNDER IMBALANCE DATA WITH SMALL BATCH SIZE

In this setting, we investigate the performance of our method on imbalanced data, with a batch size set to 1, which is common in real-world applications. Detailed results are shown in Table 1, where Sparse-CL improves by 5.3% under the SAR setting. Applying k -NL further enhances Sparse-CL performance from 63% to 65%, a 2% increase.

C.2 CAN SAR BE CLOSE TO OUR MODEL PERFORMANCE WHEN TRAINING UNDER MULTIPLE STEPS?

Because SAR cannot work under a large learning rate, we compare the strength of our method and SAR by investigating how SAR improves when adapting to the target data in multiple steps. The results in Figure 1 reveal that increasing the number of steps can improve SAR performance. However, updating the model on the sample for too long also reduces its performance (model accuracy deteriorates when the step number reaches 11). We speculate this happens due to confirmation bias, where errors accumulate after each step and significantly reduce model performance.

C.3 PERFORMANCE OF OUR METHOD WHEN STANDS ALONE

We further confirm the effect of the proposed loss compared to entropy minimization when used alone. This can be achieved by replacing the loss function in TENT (Wang et al., 2020) with ours. Similar to the previous settings, we conducted these experiments on ImageNet-C with the highest level of severity (level 5). We also consider both versions of the proposed loss: sparse-CL when used alone and in combination with k -NL. Detailed results are shown in Table 2.

C.4 RESULT ON CIFAR-10-C AND CIFAR-100-C

Besides running on different settings using ImageNet-C, we validated our method on two additional datasets: CIFAR-10-C and CIFAR-100-C under normal settings (batch size of 200). The experiments utilized 15 common corruption types (severity level 5). Generally, our loss function can outperform entropy minimization when used individually or based on SAR (combined with high entropy sample filtering and sharpness-aware loss) across all types of perturbations. More results can be found in Tables 3 and 4.

C.5 MORE RESULT ON VISDA-21

Table 5 and 6 shows the accuracy our model reach under VisDA-21 dataset (Bashkirova et al., 2022). Generally, our model acquires better performance when standing alone. Moreover, adapting this loss function under the SAR setting helps the final improvement increase up to 8%.

C.6 THE BENEFIT OF SKIPPING s HARDNESS SAMPLES

Learning with a large learning rate can be risky, especially when the training data includes noise. Even small amounts of noise can disrupt model learning under large updates. Therefore, it is crucial to carefully consider filtering out noise during the learning process. In our loss function, noise tends to come from the k -NL loss. To reduce the effect of noise from the k -NL loss (where positive samples are mistakenly considered negative), we first skip s nearby negative samples with the positive

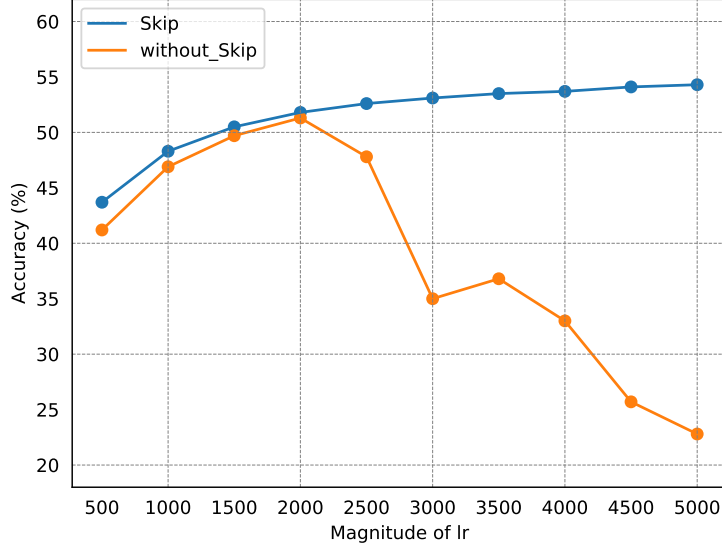


Figure 2: The changes in accuracy of our loss function with different learning rates (measured on Gaussian noise severity level 5, ImageNet-C) are significant. Learning by skipping the first s hard negative samples helps the model adapt better as we increase the learning rate, resulting in improved stability. In contrast, directly utilizing the first s highest probability samples as negative ones leads to deteriorating model performance when the learning rate is large.

Model	Running time (ms)
TENT	128
$\mathcal{L}_{sparse-CL}$	131
\mathcal{L}_{final}	133
SAR	382
$SAR^* + \mathcal{L}_{sparse-CL}$	380
$SAR^* + \mathcal{L}_{final}$	384

Table 7: The computation time of different loss functions was measured on Gaussian noise corruption type when processing one batch of data (batch size of 64) on a GPU RTX 3090. Generally, all losses yield similar computational costs (under TENT or SAR settings). Learning with sparse-CL achieves lower times, but the gap is small. This may be because we only updated the BN layers, so the difference between these losses is not significant.

one, then select consecutive k samples as negative. This section aims to highlight the advantages of this skipping procedure. Generally, Figure 2 shows that without skipping modules, the model cannot adapt to a large learning rate due to the noisy effect. On the other hand, filtering out s negative samples helps the model improve stability under large updates.¹

C.7 COMPUTATION TIME

To better understand our module, we conducted an additional experiment to measure the running time of cross-logit, k-hardness negative alone, and when combined. Table 7 shows that these two modules do not change the running time. In fact, they even help the network update faster due to sparse updating.

¹This experiment is implemented when using our loss alone instead of combining with SAR because the effect of noise could be partially mitigated by sharpness-aware loss and filtering out high entropy samples.

D RELATED WORK

D.1 DOMAIN ADAPTATION

In the era of deep learning, models are built with increasingly large sizes and trained on massive amounts of data (Dosovitskiy et al., 2020; Radford et al., 2018). These models, known as source models, are then used to transfer knowledge to target domains. This learning strategy has been a key factor in the success of deep learning models for nearly a decade when applied to real-world problems, and it is commonly known as domain adaptation (DA) (Ben-David et al., 2006; Ganin & Lempitsky, 2015). Generally, DA algorithms help improve training time and performance on the target domain by utilizing knowledge from the source domain. Currently, common techniques applied to the domain adaptation problem include unsupervised learning (Ganin & Lempitsky, 2015; Saito et al., 2018), self-supervised learning (Saito et al., 2020; Sun et al., 2019), weakly supervised learning (Inoue et al., 2018; Cozzolino et al., 2018), and feature learning (Long et al., 2018; Shen et al., 2018). The main goal of these learning techniques is to utilize the source model and data to make learning on the target domain more effective in terms of learning time and accuracy.

D.2 TEST-TIME ADAPTATION

For deployed deep learning systems, learning the target model based on traditional domain adaptation problems is no longer suitable. On the one hand, the source data is usually not accessible (for privacy reasons), so we can only take advantage of the pre-trained source model. On the other hand, the model needs to adapt and infer in an online manner, so the time for adaptation needs to be done in one or a few steps. This context-based learning is called Test-time adaptation (TTA) (Wang et al., 2020; Liu et al., 2021a). In general, learning in this direction focuses on improving the quality of the model using only target data and the source model. To update the target model quickly and efficiently, (Wang et al., 2020) points out that batch normalization layers, which work by shifting and scaling features during the learning process, could help the target model adapt well to distribution shifts. Therefore, simply fine-tuning BN layers based on entropy minimization could be efficient and save much computational cost. (Niu et al., 2022) improves the quality of TENT by filtering out low-confidence samples and using Fisher information to mitigate catastrophic forgetting. (Niu et al., 2023) enhances the model’s learning ability by adapting sharpness-aware loss and a model recovery mechanism. Additionally, (Iwasawa & Matsuo, 2021) approaches the problem based on prototype learning, where the author takes advantage of source prototypes and then updates these prototypes based on test samples. Works by (Wang et al., 2023; Li et al., 2020) leverage unsupervised learning techniques to update models based on softmax or feature spaces, while (Goyal et al., 2022; Wang et al., 2022) rely on the success of weakly supervised learning to improve model quality. Moreover, under the self-supervised learning setting, (Chen et al., 2022; Sun et al., 2020) also achieve promising results.

D.3 COMPLEMENTARY AND NEGATIVE LEARNING

Supervised learning is a popular and powerful method in machine learning and deep learning, achieving impressive results in various tasks. However, it requires large amounts of labeled data, which can be costly and difficult to obtain, especially for complex problems such as segmentation. Moreover, the quality of the labels can affect the performance of the learning models, as noise and errors can be introduced during the labeling process. To address these challenges, (Ishida et al., 2017) proposed complementary learning, a method that leverages information from other classes (complementary classes) besides the ordinary class. (Ishida et al., 2019; Yu et al., 2018) extended this framework to different loss functions and provided theoretical guarantees. However, these methods assume that the labels are clean and accurate, and cannot handle noisy labels. To overcome this limitation, (Kim et al., 2019) proposed a novel algorithm called negative learning, which can learn from both ordinary labels and negative labels (labels that are opposite to the true labels). They empirically show that negative learning can improve the robustness and accuracy of learning models under noisy label settings.

D.4 STABILITY LEARNING

Deep learning models have achieved remarkable success in many practical applications (He et al., 2016; Brown et al., 2020), but they also suffer from instability issues when dealing with challenging real-world problems, such as noisy data (Natarajan et al., 2013), imbalanced data (Haixiang et al., 2017), or adversarial attacks (Goodfellow et al., 2014). These issues can degrade the quality of the models during training and inference, affecting their robustness and generalization. To address these challenges, various research directions have been proposed to enhance the stability of neural network training in different contexts, such as learning with noisy labels (Song et al., 2022), imbalanced learning (Fernández et al., 2018), or adversarial learning (Zhang et al., 2018). Moreover, many studies have shown that the loss landscape plays a crucial role in the generalization of neural networks, as it reflects the complexity and diversity of the solutions (Li et al., 2018; Wu et al., 2020). Therefore, some methods have been developed to optimize the loss landscape and find more stable regions for the models (Foret et al., 2020; Kwon et al., 2021). For example, (Foret et al., 2020) proposed a sharpness-aware minimization method that simultaneously minimizes the loss value and the loss sharpness, leading to better generalization and robustness. Besides the loss landscape, the gradient norm during training also indicates the stability of the network, as stable networks tend to have smaller gradient variance (less fluctuation of the gradient norm across different batches of data) (Johnson & Zhang, 2013; Liu et al., 2021b). Based on this idea, some works have introduced methods to improve network learning efficiency based on gradient variance. For example, (Faghri et al., 2020) proposed a gradient clustering method that reduces gradient variance by using stratified sampling.

REFERENCES

- Maksym Andriushchenko, Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. Sgd with large step sizes learns sparse features. In *International Conference on Machine Learning*, pp. 903–925. PMLR, 2023.
- Dina Bashkirova, Dan Hendrycks, Donghyun Kim, Haojin Liao, Samarth Mishra, Chandramouli Rajagopalan, Kate Saenko, Kuniaki Saito, Burhan Ul Tayyab, Piotr Teterwak, et al. Visda-2021 competition: Universal domain adaptation to improve performance on out-of-distribution data. In *NeurIPS 2021 Competitions and Demonstrations Track*, pp. 66–79. PMLR, 2022.
- Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.
- Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.
- Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv preprint arXiv:1812.02510*, 2018.
- James Diffenderfer, Brian Bartoldson, Shreya Chaganti, Jize Zhang, and Bhavya Kailkhura. A winning hand: Compressing deep networks can improve out-of-distribution robustness. *Advances in neural information processing systems*, 34:664–676, 2021.
- 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.

-
- 378 Fartash Faghri, David Duvenaud, David J Fleet, and Jimmy Ba. A study of gradient variance in deep
379 learning. *arXiv preprint arXiv:2007.04532*, 2020.
- 380
381 Lei Feng, Takuo Kaneko, Bo Han, Gang Niu, Bo An, and Masashi Sugiyama. Learning with mul-
382 tiple complementary labels. In *International Conference on Machine Learning*, pp. 3072–3081.
383 PMLR, 2020.
- 384 Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Fran-
385 cisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.
- 386
387 Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimiza-
388 tion for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.
- 389 Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In
390 *International conference on machine learning*, pp. 1180–1189. PMLR, 2015.
- 391
392 Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial
393 examples. *arXiv preprint arXiv:1412.6572*, 2014.
- 394 Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and Zico Kolter. Test-time adaptation via conjugate
395 pseudo-labels. *arXiv preprint arXiv:2207.09640*, 2022.
- 396
397 Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning
398 from class-imbalanced data: Review of methods and applications. *Expert systems with applica-*
399 *tions*, 73:220–239, 2017.
- 400 Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recog-
401 nition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.
402 770–778, 2016.
- 403
404 Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep
405 learning: Pruning and growth for efficient inference and training in neural networks. *The Journal*
406 *of Machine Learning Research*, 22(1):10882–11005, 2021.
- 407
408 Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-
409 supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE*
410 *conference on computer vision and pattern recognition*, pp. 5001–5009, 2018.
- 411
412 Eugenia Iofinova, Alexandra Peste, Mark Kurtz, and Dan Alistarh. How well do sparse imagenet
413 models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
414 *Recognition*, pp. 12266–12276, 2022.
- 415
416 Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary
417 labels. *Advances in neural information processing systems*, 30, 2017.
- 418
419 Takashi Ishida, Gang Niu, Aditya Menon, and Masashi Sugiyama. Complementary-label learning
420 for arbitrary losses and models. In *International Conference on Machine Learning*, pp. 2971–
421 2980. PMLR, 2019.
- 422
423 Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic do-
424 main generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.
- 425
426 Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance
427 reduction. *Advances in neural information processing systems*, 26, 2013.
- 428
429 Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy
430 labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 101–
431 110, 2019.
- Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot
image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.
- Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. Asam: Adaptive sharpness-
aware minimization for scale-invariant learning of deep neural networks. In *International Con-*
ference on Machine Learning, pp. 5905–5914. PMLR, 2021.

432 Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss land-
433 scape of neural nets. *Advances in neural information processing systems*, 31, 2018.

434

435 Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised
436 domain adaptation without source data. In *Proceedings of the IEEE/CVF conference on computer
437 vision and pattern recognition*, pp. 9641–9650, 2020.

438

439 Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexan-
440 dre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural
441 Information Processing Systems*, 34:21808–21820, 2021a.

442

443 Yuxiang Liu, Jidong Ge, Chuanyi Li, and Jie Gui. Delving into variance transmission and nor-
444 malization: Shift of average gradient makes the network collapse. In *Proceedings of the AAAI
445 Conference on Artificial Intelligence*, volume 35, pp. 2216–2224, 2021b.

446

447 Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Transferable
448 representation learning with deep adaptation networks. *IEEE transactions on pattern analysis
449 and machine intelligence*, 41(12):3071–3085, 2018.

450

451 Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with
452 noisy labels. *Advances in neural information processing systems*, 26, 2013.

453

454 Shuaicheng Niu, Jiayang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui
455 Tan. Efficient test-time model adaptation without forgetting. In *International conference on
456 machine learning*, pp. 16888–16905. PMLR, 2022.

457

458 Shuaicheng Niu, Jiayang Wu, Yifan Zhang, Zhiqian Wen, Yaofo Chen, Peilin Zhao, and
459 Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint
460 arXiv:2302.12400*, 2023.

461

462 Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language under-
463 standing by generative pre-training. 2018.

464

465 Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier dis-
466 crepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on com-
467 puter vision and pattern recognition*, pp. 3723–3732, 2018.

468

469 Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation
470 through self supervision. *Advances in neural information processing systems*, 33:16282–16292,
471 2020.

472

473 Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face
474 recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern
475 recognition*, pp. 815–823, 2015.

476

477 Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation
478 learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*,
479 volume 32, 2018.

480

481 Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in
482 neural information processing systems*, 29, 2016.

483

484 Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy
485 labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning
486 Systems*, 2022.

487

488 Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through
489 self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

490

491 Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time train-
492 ing with self-supervision for generalization under distribution shifts. In *International conference
493 on machine learning*, pp. 9229–9248. PMLR, 2020.

486 Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully
487 test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.
488

489 Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation.
490 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.
491 7201–7211, 2022.

492 Shuai Wang, Daoan Zhang, Zipei Yan, Jianguo Zhang, and Rui Li. Feature alignment and uniformity
493 for test time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*
494 *Pattern Recognition*, pp. 20050–20060, 2023.

495 Ross Wightman. Pytorch image models. [https://github.com/rwightman/](https://github.com/rwightman/pytorch-image-models)
496 [pytorch-image-models](https://github.com/rwightman/pytorch-image-models), 2019.
497

498 Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust gener-
499 alization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

500 Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary
501 labels. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 68–83, 2018.
502

503 Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adver-
504 sarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp.
505 335–340, 2018.
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539