Leveraging Test-Time Consensus Prediction for Robustness against Unseen Noise

Anindya Sarkar [*], Anirban Sarkar [*], Vineeth N Balasubramanian
Indian Institute of Technology, Hyderabad
anindyasarkar.ece@gmail.com, cs16resch11006@iith.ac.in, vineethnb@iith.ac.in

Abstract

We propose a method to improve DNN robustness against unseen noisy corruptions, such as Gaussian noise, Shot Noise, Impulse Noise, Speckle noise with different levels of severity by leveraging ensemble technique through a consensus based prediction method using self-supervised learning at inference time. We also propose to enhance the model training by considering other aspects of the issue i.e. noise in data and better representation learning which shows even better generalization performance with the consensus based prediction strategy. We report results of each noisy corruption on the standard CIFAR10-C and ImageNet-C benchmark which shows significant boost in performance over previous methods. We also introduce results for MNIST-C and TinyImagenet-C to show usefulness of our method across datasets of different complexities to provide robustness against unseen noise. We show results with different architectures to validate our method against other baseline methods, and also conduct experiments to show the usefulness of each part of our method.

1. Introduction

Generalization performance of Deep Neural Networks (DNNs) is a very important objective, as the networks are susceptible to fail against noise at test time. In recent years, researchers have shown many examples of this kind [10], which raises serious questions about deployment of the seemingly good DNN models in the wild. Although these issues were shown with known types of noise, this problem is actually more challenging because it is difficult to predict what noise will occur at test time.

Recent years have seen a few different efforts in developing models robust to unseen noise. Adversarial joint training [11] was one of the early efforts that focused on improving model robustness by enhancing model generalizability. In particular, this method attempted to find a robust model against noise by adversarial training with a supervised head attached to the model. Though this method performed better than a naturally trained model, there still existed a big gap in performance between test accuracy on clean data and noisy data. More recently, [27] proposed a test-time training method to improve model generalization which showed performance boost over [11] on noisy data. A more recent work [1] further showed improvement over such test-time training [27]. Augmentation-based methods, on the other hand [2, 7, 12, 24] attempt to address model generalization by carefully augmenting the training dataset through different means. However, augmentation methods require a significant increase in the size of the training dataset, and are also known to fail when the test-time distribution differs from the distribution of augmented training data [18]. In this work, we seek to propose a method that can address model robustness to unseen noise by only training on clean data (no additional training data including augmentations).

To this end, we take advantage of ensembled inference through a novel test-time consensus-based prediction method that allows for better generalization at inference. We show that such an approach shows excellent performance against unseen noise, when compared with aforementioned state-of-the-art baselines, especially when no additional data beyond the clean training data is used for training. Building on [27], this method leverages a self-supervision pretext task at test-time to iteratively update the model and predict the class label as a majority vote over multiple predictions of updated models at inference. We call this Test-Time Consensus Prediction (TTCP). In order to further improve model performance against unseen noise, we also propose an extended framework (TTCP++), where a training phase is introduced to: (i) retrieve the latent data manifold from clean data using the idea of quantized latents [28]; and (ii) improve the representations learned by the backbone network used in TTCP via knowledge distillation from a pre-trained teacher network. When quantized latents are used, although the reconstructed images are less noisy, due to the discrete nature of the latent space, the method fails to preserve local texture details during image reconstruction. We leverage knowledge distillation from a pre-trained teacher to help the backbone network learn better representations from such reconstructed images. Our results on multiple benchmark datasets show significant improvement over existing methods, corroborating our claim.

[*]equal contribution

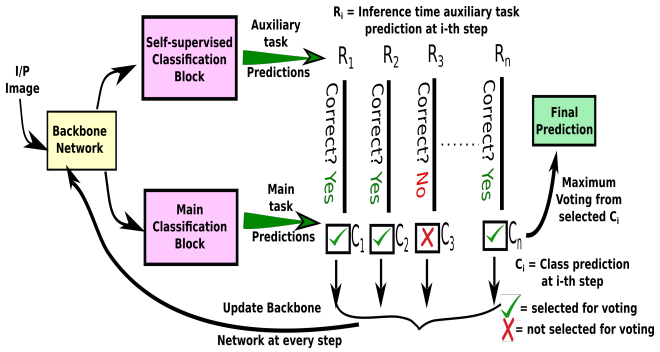

Figure 1. Proposed Test-Time Consensus Prediction (TTCP) method

Our key contributions are as follows:

- We propose a novel test-time consensus prediction (TTCP) strategy to achieve better model robustness through improved generalization performance against unseen noise.
- We propose an extended framework, TTCP++, to exploit quantized latents and knowledge distillation in a training phase, to boost the performance of the proposed TTCP method on unseen noise.
- Our results on CIFAR10-C and ImageNet-C are a significant improvement over previous methods based on improved training. We also studied our method on MNIST-C and TinyImagenet-C datasets, which are the first results in this context, and report strong results here too.
- We perform consistently against all kinds of noise on CIFAR10-C and ImageNet-C datasets compared to augmentation-based methods, even without the use of any augmentation and training only on clean data.
- Detailed ablation studies are presented showing the usefulness of each component of our overall TTCP++ framework.

We discuss earlier efforts which are based on different related perspectives, in Sec A which is deferred to Appendix due to space constraints.

2. Proposed Methodology

We now introduce our Test-Time Consensus Prediction method (TTCP) towards achieving improved robustness against unseen noise. We then present the extension of our method to TTCP++ which improves the training procedure to further get improvements on noisy datasets.

2.1. Test-Time Consensus Prediction (TTCP)

Given training data $\{(x_i, y_i), i = 1, \cdots, n\}$ drawn i.i.d. from a distribution P and model parameters θ, we consider the classification task loss function $\mathcal{L}_m(x_i, y_i, \theta)$ as the main task. We leverage the fact that self-supervised learning empowers model training by improving the intermediate representation with better semantic meaning, which may be beneficial to a downstream task. The labels for a self-supervised task can be generated for free, and a corresponding supervised loss is computed based on the task. We refer to the self-supervised task as *auxiliary task* which yields the loss $\mathcal{L}_{ss}(x_i, y_i, \tilde{\theta})$.

Algorithm 1: Test-Time Consensus Prediction

Input: Test sample x, Self-supervision head f parametrized by θ_s, Classification head g parametrized by θ_m, Pretrained backbone network e parametrized by θ_e, Ground truth auxiliary task output in i^{th} step R_i^*, Operator $\phi(\cdot)$ which converts a softmax output to a one-hot vector, Number of classes for main classification task L

Output: Predicted class label for test sample x

Initialize **vote counter**. f = $f_{pretrained}$; e = $e_{pretrained}$;
for $i = 1, 2, ..., M$ **do**
  x_{ss}= $T(x)$; *(Random auxiliary transformation)*
  Compute auxiliary task prediction $\hat{R}_i = f(e(x_{ss}))$
  Compute main classification task prediction $\hat{y}_i = g(e(x_{ss}))$
  Compute classwise vote $V_i = \mathbb{I}[\hat{R}_i = R_i^*]\ \phi(\hat{y}_i)$
  Update θ_e
end
Predict class label $\arg\max_{j \in L}(\sum_{i=1}^{M}(V_i))$

Now, consider a (Y-shaped) DNN represented as a backbone network with two task-specific heads – one for the main classification task and the other for the auxiliary task. Let the model parameters for the backbone network be θ_e, the main task head be θ_m and the auxiliary task head be θ_s, i.e. $\theta = (\theta_e, \theta_m)$ and $\tilde{\theta} = (\theta_e, \theta_s)$. This DNN is trained by minimizing the loss terms for both tasks, l_m and l_{ss}. Assuming the model parameters θ_e and θ_m are already trained using a prior training procedure, following [27], we focus our efforts on the test-time (or inference stage). Given a test data point x forward-propagated through the abovementioned DNN, a gradient step is taken with the objective of minimizing the auxiliary task loss on x i.e.

$$\min_{\theta_e} \mathcal{L}_{ss}(x, \theta_e, \theta_s) \tag{1}$$

For such a gradient step which updates the shared backbone network parameters θ_e^*, the model predicts a class label (main task) through the parameters (θ_e^*, θ_m).

Leverage the capabilities of self-supervision and shared parameter updation at test-time, an auxiliary transformation (e.g. rotation, which defines a corresponding self-supervised task – rotation prediction in this case) is applied to x and passed through the network. At the i^{th} such test-time step, we observe both the prediction of the auxiliary task as well as the main task before the gradient step is taken to update the shared backbone parameters. Let R_i^* denoted the true label for the auxiliary task in the i^{th} step and $\hat{R}_i$ denote the predicted auxiliary task output in the same step.

This step is performed a pre-defined number of times, each time with a different auxiliary transformation applied to x under the same self-supervised task (different rotation angle, for example).

At the completion of these steps, we define our final output of test-time consensus prediction as:

$$\arg\max_{j \in L}(\sum_{i} \mathbb{I}[\hat{R}_i = R_i^*]\ \phi(\hat{y}_i)) \tag{2}$$

where $\phi(.)$ returns a one-hot vector given a softmax output (with the winning position denoted by a 1 and rest zeros), $\mathbb{I}$ is the indicator function, L is the number of classes for the main classification task, $\hat{y}$ is the predicted softmax output of the main classification task, and $C_i = \arg\max_{j \in L}(\hat{y}_i)$. In words, Eqn 2 states that our final output is the consensus of predictions on the main classification output for every transformed input where the auxiliary task head predicts the correct expected output. We term this complete strategy Test-Time Consensus Prediction (TTCP). The TTCP method is summarized in Fig 1, and described in Algorithm 1. We note that similar to [27], only a few steps are required at test-time thereby resulting in minimal computational overhead (described further in Sec B).

2.2. Beyond Test-Time Consensus Prediction (TTCP++)

While TTCP focuses only on test-time, we observe that improvements in obtaining a more robust representation of input data can further help TTCP. To this end, we include a training phase procedure for handling unknown noise at test-time. In particular, we propose a two-staged approach: (i) retrieve the latent data manifold from given clean training data; and (ii) improve the feature representations of the backbone network using knowledge distillation. We now describe each of these steps that together comprise TTCP++.

Retrieving the latent data manifold: Removing noise from data has long been an important topic of research. It is generally hypothesized that a data point x and its noisy version x_{noise} are arbitrarily close on the true data manifold. One of the well-known approaches to retrieve this true data manifold is to use a denoising autoencoder (DA) [29]. Such an approach attempts to remove noise by generating clean images from its noisy version, and works if the type of noise is known beforehand. However, such an approach does not work well (shown in our ablation studies) when handling unknown or unseen noise, given only clean data at training time. We hence instead propose to leverage the idea of quantized latents to mitigate noise in data. Discretization inherently makes two different but nearby points from the given data distribution map to the same point or bring them closer in the latent data manifold. We follow a vector quantization-based approach [28] to achieve this objective.

Our model consists of a standard convolutional encoder-decoder architecture with an intermediate vector quantization (VQ) layer which takes care of building a discrete latent space. More specifically, the encoder network, Z_e, is a fully convolutional neural network which maps input images to an output feature map of size $w \times h \times d$. This provides $w \times h$ d-dimensional vectors, each of which is mapped to one latent code from a set of k learned latent codes $\{e_1, \cdots, e_k\} \in \mathbb{R}^d$ through a mapping (VQ) layer. This is achieved by minimizing the L_2-distance between each of the d-dimensional vectors (obtained as output of Z_e) with the latent codes e_is, i.e.

$$\hat{Z}_e(p,q,:) = e_j, \text{ where } j = \arg\min_{i \in \{1,\cdots,k\}} ||Z_e(p,q,:) - e_i||_2 \tag{3}$$

where Z_e denotes $Z_e(x)$ for a given input x, p and q are indices over $w \times h$ d-dimensional vectors. The output of the encoder hence is $\hat{Z}_e(x)$, a $w \times h \times d$ feature map, where the depth vector at each pixel is replaced by the nearest latent code vector. $\hat{Z}_e(x)$ is then provided as input to a fully convolutional decoder, Z_d, which attempts to reconstruct the input image.

Due to the discrete bottleneck layer in between, the training of this model is not straightforward. The training objective includes the reconstruction loss (mean squared error) and two VQ-based loss terms which guarantee that encoder outputs stay close to the embedding vector entries they are matched to, as below:

$$\arg\min_{\hat{Z}_e, Z_d, \{e_i\}} \log p\big(\tilde{x}|\hat{Z}_e(x)\big) + ||Z_e(x) - \overline{\hat{Z}_e(x)}||^2 + ||\overline{Z_e(x)} - \hat{Z}_e(x)||^2 \tag{4}$$

where $\overline{\cdot\cdot}$ denotes the stop gradient operation, i.e., during forward pass, this corresponds to the identity, but during back-propagation, no gradients flow through this operation. We follow [28] for the rest of the training procedure.

Importantly, we use clean training data to learn vector quantized latents and show that the trained model works well with unseen noise at test-time. We find that this model is capable of removing local textures (including noise), but keeps the global content of an image while reconstructing the image. (We note that this step is offline, and can be done prior to the training of the backbone network used in TTCP.) We present sample visualizations in Figs 3, 5 and 6 for different noises on images from CIFAR10-C, MNIST-C and Tiny-ImageNet-C dataset to show the potential of this method.

Learning robust representations: While the above step captures global details of the original image, it also misses texture details of original image (as shown in Figs 3). Due to this lossy nature of the reconstructed images, we noticed that the models, trained with these images – while handling unseen noise – report a drop in clean test accuracy ($\sim$9–10%) compared to a model trained on normal data.

To address this issue, we consider a pretrained (on clean data) Teacher Network. Our objective herein is to lever-

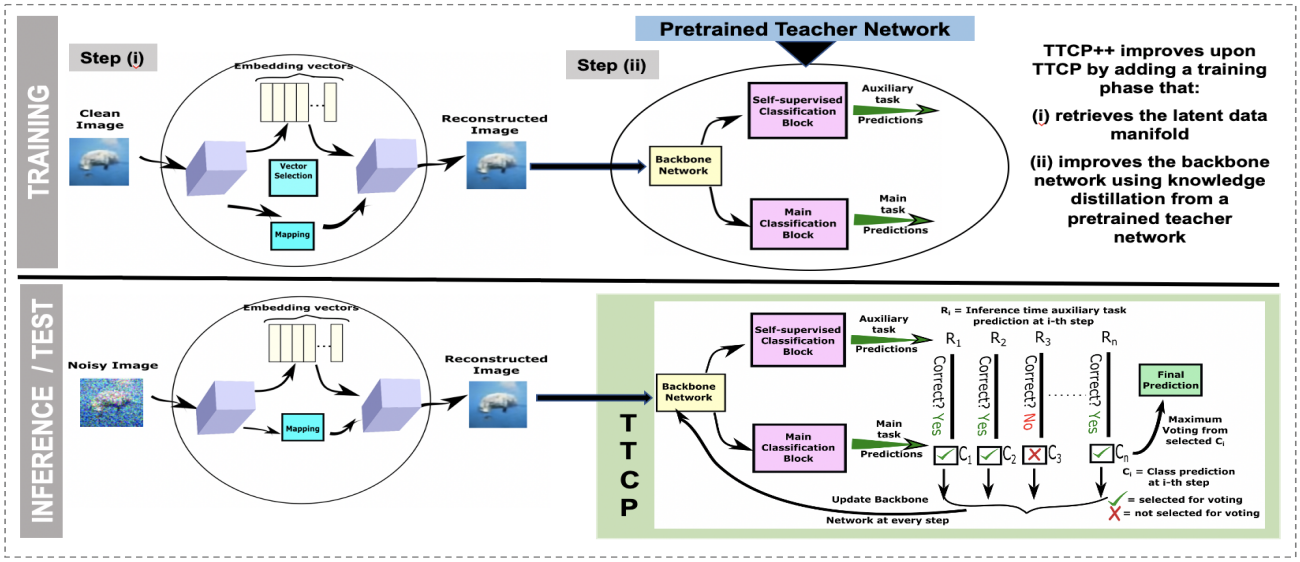

Figure 2. Overall framework of TTCP++

age the guidance of the teacher network to help the student model in learning refined representations from images only with the global content obtained as output of the quantized latent (VQ) step above. The said purpose is achieved by incorporating the following objective:

$$\mathcal{L}_{KD} = ||\text{logit}(T(x)) - \text{logit}(S(x_{recons}))||_2 \quad (5)$$

where $T(\cdot)$ denotes the pretrained teacher network, $S(\cdot)$ denotes our backbone network from TTCP (student network in this context), logit$(\cdot)$ represents the logits of the corresponding network, and x_{recons} denotes the reconstructed image obtained from the previous step when x is provided as input. The student network architecture here follows a Y-shape structure (similar to TTCP, for later use of TTCP at test-time) and contains a *self-supervision head* and a *classification head* for auxiliary task prediction and classification respectively. Both these heads follow a shared backbone network. Altogether, the student network is trained with standard classification loss (cross-entropy loss), self-supervision loss (auxiliary prediction loss) and logit similarity loss (minimizing the L_2 distance of logits generated by teacher and student), as given below:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{SS} + \mathcal{L}_{KD} \quad (6)$$

where $\mathcal{L}_{CE}$ denotes cross entropy loss and $\mathcal{L}_{SS}$ denotes self supervision loss, as before in Sec 2.1.

Inference/Test-time: After the training phase, during inference, we input the test data through the VQ module to obtain the reconstructed image. This is then input to the TTCP module to obtain the final prediction as in Sec 2.1. We name this extension TTCP++ when we consider vector quantization and a pretrained teacher network for model training along with TTCP during inference time. Adding this training phase helps improve clean accuracy by $\sim$3%. Additionally, with this training phase, TTCP during inference phase also achieves improvements on clean test accuracy ($\sim$3.5%) as well as improvements on unseen noisy test data (ranging from 6-10% across different noise).

We conducted a comprehensive suite of experiments which are discussed in detail in Sec B of Appendix, which shows promising improvement in performance over baseline methods. Ablation studies of individual components and other combinations are studied in Sec C of Appendix which justifies importance of all these components. We report results with our method on CIFAR10-C, Tiny-ImageNet-C, ImageNet-C and MNIST-C datasets [10], going beyond earlier related methods – joint training (JT) [11], test-time training (TTT) [27] and SSDN [1] – which focused on CIFAR10-C.

3. Conclusions and Future Work

Most real-world environments are inherently noisy, thus hindering DNN models from being deployed in real-world applications, especially in in-the-wild with unknown or unseen noise. In this work, we propose a simple yet effective test-time consensus prediction (TTCP) approach that addresses model robustness to robust noise with training only on clean data. We further propose an extended version, TTCP++, where we add a training phase based on quantized latents and knowledge distillation, that helps improve the performance of TTCP further on unseen noise. Our comprehensive results on multiple benchmark datasets against state-of-the-art baselines show significant promise in using our approach for deploying DNN models in in-the-wild settings with unseen noise.

References

[1] Tomer Cohen, Noy Shulman, Hai Morgenstern, Roey Mechrez, and Erez Farhan. Self-supervised dynamic networks for covariate shift robustness. *arXiv preprint arXiv:2006.03952*, 2020.

[2] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *Proceed-ings of IEEE Conference on Computer Vision and PatternRecognition*, 2018.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[5] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

[6] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017.

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *International Conference on Learning Representations*, 2018.

[8] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[10] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[11] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pages 15663–15674, 2019.

[12] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *Advances in Neural Information Processing Systems*, 2019.

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

[14] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

[18] Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. *arXiv preprint arXiv:2102.11273*, 2021.

[19] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[20] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[21] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[22] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

[23] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[24] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. *European Conference on Computer Vision*, 2020.

[25] Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Knowledge distillation beyond model compression. *arXiv preprint arXiv:2007.01922*, 2020.

[26] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33, 2020.

[27] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning (ICML)*, 2020.

[28] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

[29] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust

features with denoising autoencoders. In *International Conference on Machine Learning*, page 1096–1103, 2008.

[30] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

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

[32] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015.

[33] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8052–8060, 2018.

[34] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[35] Han Yang, Xiao Yan, Xinyan Dai, and James Cheng. Self-enhanced gnn: Improving graph neural networks using model outputs. *arXiv preprint arXiv:2002.07518*, 2020.

[36] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[37] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[38] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.