INVESTIGATING GENERALIZATION IN NEURAL NETWORKS UNDER OPTIMALLY EVOLVED TRAINING PERTURBATIONS

Subhajit Chaudhury[⋆†] *Toshihiko Yamasaki*[⋆]
{subhajit, yamasaki}@hal.t.u-tokyo.ac.jp

[⋆] The University of Tokyo [†]IBM Research AI - Tokyo

ABSTRACT

In this paper, we study the generalization properties of neural networks under input perturbations and show that minimal training data corruption by a few pixel modifications can cause drastic overfitting. We propose an evolutionary algorithm to search for optimal pixel perturbations using novel cost function inspired from literature in domain adaptation that explicitly maximizes the generalization gap and domain divergence between clean and corrupted images. Our method outperforms previous pixel-based data distribution shift methods on state-of-the-art Convolutional Neural Networks (CNNs) architectures. Interestingly, we find that the choice of optimization plays an important role in generalization robustness due to the empirical observation that SGD is resilient to such training data corruption unlike adaptive optimization techniques (ADAM).

Index Terms— Generalization in deep learning, data poisoning, adaptive optimization, data distribution shift

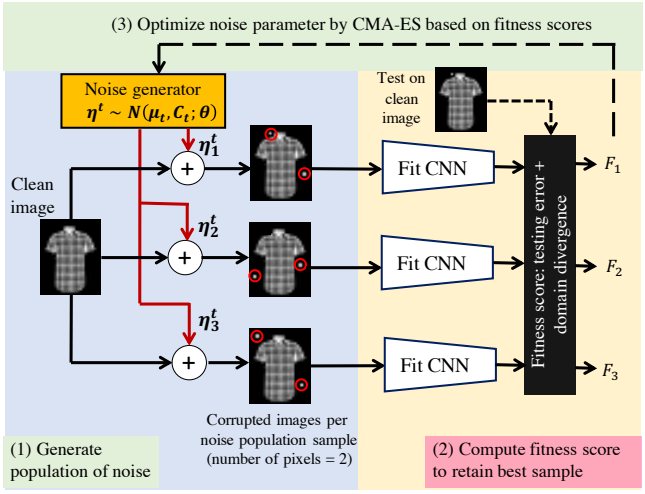

Fig. 1: Overview of our proposed noise optimization algorithm

1. INTRODUCTION

Deep learning has shown notable empirical success in various application areas. Typically, in an over-parametrized setting with a highly non-convex loss surface, classical learning theory [1] predicts that deep neural networks should have a high out-of-sample error because the solution is likely to get stuck at a local minimum. Nonetheless, deep neural networks appear to generalize well even in small data regimes. Numerous recent works have sought to explain generalization in neural networks. Zhang et al. [2] showed that neural networks can fit random noise and labels, thus refuting the finite sample expressivity argument. Another view [3] as to why neural networks generalize well, studies the loss surface geometry around the learned parameter and shows that sharper minima solutions tend to generalize poorly compared to flatter minima which were contested by Dinh et al. [4]. Some recent research [5, 6] also demonstrates that vanilla SGD optimization has better generalization ability than adaptive optimization methods.

Our method is similar to *Adversarial Distribution Shift* (ADS) presented in [7] where benign perturbations are added to the training data causing neural networks to learn task-irrelevant features. Specifically, [7] studied the effect of single-pixel perturbations on MNIST training images on clean test performance. Data poisoning attacks [8, 9, 10] are also related to such an approach where the adversary injects a few malicious samples in the training data to cause incorrect classification (typically targeted) during inference. Tanay et al. [11] showed that neural network models can be made almost arbitrarily sensitive to a single-pixel while maintaining identical test performance between models. However, poisoning methods [9, 12, 13] usually modify some part of the decision boundary by adding malicious training samples for targeted misclassifications, which is different from our approach of optimal ADS. Moreover, our motivation in this work is to analyze how optimization methods, specifically adaptive and non-adaptive algorithms, contribute to generalization robustness which is different from the typical objective of data poisoning methods.

In this paper, we find optimal training ADS that cause a high generalization gap between corrupted training and clean images during inference while limiting the attack to a few pixels only. The overview of our method is shown in Figure 1.Our contribution in this paper is two-fold. Firstly, we propose a novel fitness function for the CMA-ES algorithm to find op-

timal pixel disturbance, using domain adaptation theory. Our method outperforms previous heuristic ADS method presented in [7]. Secondly, our analysis reveals that the choice of optimization technique plays an important role in generalization robustness. Specifically, vanilla SGD is found to be surprisingly resilient against training sample perturbations compared to adaptive optimization methods like ADAM, which calls into question the effectiveness of such popular adaptive optimization methods towards generalization robustness.

2. PROBLEM SETUP

We consider a multi-class classification task with input space $X \in \mathbb{R}^N$ and label space $Y = \{1, ..., N_c\}$. The true data distribution is given as, $S = \{\boldsymbol{x}_i, y_i\}_{i=1}^n \sim \mathcal{D}_S$. Our goal is to train a classifier on a perturbed version of the true data samples such that the empirical risk (or test error) on the true uncorrupted samples is maximized. Considering that for each sample in S, we can draw class-wise input perturbations, $\delta = \{\boldsymbol{\eta}_i\}_{i=1}^{N_c} \sim N(\boldsymbol{m}, \boldsymbol{\Sigma})$, parameterized by the mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$, which are added to the true samples, $\boldsymbol{x}_i^p = \boldsymbol{x}_i + \boldsymbol{\eta}_{y_i}$, where noise encoding each class information is added to training images. The joint distribution of the perturbed data, constructed by assigning labels of the true samples to the corresponding perturbed samples, given as $P = \{\boldsymbol{x}_i^p, y_i^p\}_{i=1}^n \sim \mathcal{D}_P$. In this paper, we work with image inputs and perturb a few pixels to analyze generalization sensitivity to small changes in training inputs.

Let us define a classifier function $h : X \rightarrow Y$ from a hypothesis space $\mathcal{H}$. The corresponding empirical risk on samples drawn from a distribution $\mathcal{D}$ is defined as, $R_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big(I[h(\boldsymbol{x}) \neq y]\big)$, which signifies the error on the samples drawn from $\mathcal{D}$. Our objective is to find optimal perturbation parameter that increases the empirical risk on the clean samples while minimizing it on the corrupted samples, thus compromising generalization in neural networks, given as

$$\max_{\boldsymbol{m},\boldsymbol{\Sigma}} \Big(R_{\mathcal{D}_S}(h^*) - R_{\mathcal{D}_P}(h^*) \Big) \; s.t. \; h^* = \arg\min_{h\in\mathcal{H}} R_{\mathcal{D}_P}(h). \tag{1}$$

The above objective finds optimal perturbation parameter that increases the empirical risk on the clean samples while minimizing it on the corrupted samples, thus compromising generalization in neural networks.

3. MAXIMUM DOMAIN DIVERGENCE BASED EVOLUTIONARY STRATEGY (MDD-ES)

The objective function in Equation 1 requires a nested minimization for classifier training and empirical risk maximization for optimal noise search. This presents difficulty in using standard gradient-based optimization methods for searching the optimal pixel perturbations. Therefore, we use a black-box optimization technique, specifically Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [14], which has been shown to work well in high-dimensional problems [15]. However, simply using empirical risk (generalization gap) measure on clean samples as a fitness score might require more generations for convergence. However, each generation of the CMA-ES is computationally expensive (due to multiple CNN training rounds). Therefore, we propose a novel fitness score inspired by the domain divergence literature that provides an additional signal for convergence, leading to improved noise optimization properties from fewer generations.

3.1. Measuring Domain-Divergence

Considering a domain $\mathcal{X}$ and a collection of subsets of $\mathcal{X}$ as $\mathcal{A}$. Given two domain distributions D_S and D_T over $\mathcal{X}$, and a hypothesis class $\mathcal{H}$, Shai et al. [16, 17] showed that domain divergence ($\mathcal{H}$-divergence) for the hypothesis space of linear classifiers can be approximately computed by the empirical $\mathcal{H}$-divergence from samples $\boldsymbol{x}_i^s \sim \tilde{D}_S$ and $\boldsymbol{x}_i^t \sim \tilde{D}_T$ as,

$$\hat{d}_{\mathcal{H}}(S,T) \stackrel{\text{def}}{=} 2\Bigg(1 - \min_{h\in\mathcal{H}} \Bigg[\frac{1}{n}\sum_{i=1}^{n} I[h(\boldsymbol{x}_i^s)=0] + \frac{1}{n'}\sum_{i=n+1}^{N} I[h(\boldsymbol{x}_i^t)=1]\Bigg]\Bigg), \tag{2}$$

where n samples from the source domain and n' samples from the target domain is drawn. The proxy $\mathcal{A}$-distance is computed as, $\hat{d}_{\mathcal{A}} = 2(1 - 2\epsilon)$ according to [16], where ϵ is the discriminator error.

3.2. Bound on Target Risk

We are interested in finding a bound of the target empirical risk obtained by learning a classifier of the source samples. Shai et al. (and later used by Ganin et al. [16, 17, 18]) showed that the bound on target risk can be computed in terms of the proxy $\mathcal{A}$-distance defined above, as follows,

Theorem 1. Considering $\mathcal{H}$ be a hypothesis class of VC dimension d, for n samples $S \sim (\tilde{D}_S)^n$ and $T\tilde{(}\hat{D}_T)^n$, then with probability $1 - \delta$ over the choice of samples, for every $h \in \mathcal{H}$:

$$\hat{R}_T(h) \leq \hat{R}_S(h) + \sqrt{\frac{4}{n}\left(d\log\tfrac{2e\,n}{d} + \log\tfrac{4}{\delta}\right)} + \hat{d}_{\mathcal{H}}(S,T) + 4\sqrt{\frac{1}{n}\left(d\log\tfrac{2n}{d} + \log\tfrac{4}{\delta}\right)} + \beta, \tag{3}$$

with $\beta \geq \inf_{h^*\in\mathcal{H}} [R_S(h^*) + R_T(h^*)]$ and $\hat{R}_S(h)$ is the empirical source risk.

Given a fixed hypothesis space, we observe that increasing the $\mathcal{H}$-divergence between the two domains would make the above bound loose. Since we are interested in maximizing the

target risk, pixel perturbations that increase the $\mathcal{H}$-divergence between corrupted and clean data would be more likely to fool the neural network. We use this insight to craft a fitness score that favors solutions with high domain divergence between the clean and perturbed distributions.

3.3. Proposed Fitness Score based CMA-ES Optimization

Using the insights developed in the previous section, we propose MDD-ES algorithm that utilizes a fitness score measuring, (i) semantic mismatch score, (ii) domain divergence score. Given training data, $(x, y) \sim \mathcal{D}$, and initial CMA-ES parameters, $\boldsymbol{m}_0, \boldsymbol{\Sigma}_0, \sigma_0$, we sample a population of noise for each generation, $\{\delta_j\}_{j=1}^{\lambda} \sim N(\boldsymbol{m}_t, \boldsymbol{\Sigma}_t)$. For each sample in the current generation t, we obtain the optimal weights, $\boldsymbol{\theta}^*$, by training a CNN (F_θ^j) from scratch on the corrupted training samples $\{x + \delta_j\}$. We compute the semantic mismatch score for the j^{th} noise sample as $\mathcal{F}_m^j = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[l_{CE}(F_\theta^j(x + \delta_j), y) - l_{CE}(F_\theta^j(x), y)\big]$, where l_{CE} is the cross-entropy loss. This score encourages high loss of generalization between clean and corrupted samples drawn from the training distribution. To obtain the domain divergence score, we train a discriminator with corrupted samples as label 0 and clean samples as label 1. The domain divergence score is computed as, $F_d^j = (1 - 2\epsilon)$, where ϵ is the error of the trained discriminator. The overall fitness score for the CMA-ES algorithm is computed as the combination of above score, $\mathcal{F}_j = \mathcal{F}_m^j + \mathcal{F}_d^j$. After each generation, the sampling parameters are updated by the CMA-ES algorithm to favor the pixel perturbations corresponding to the top-performing fitness scores, $\boldsymbol{m}_{t+1}, \boldsymbol{\Sigma}_{t+1}, \sigma_{t+1} = \text{CMA-ES}(\boldsymbol{m}_t, \boldsymbol{\Sigma}_t, \sigma_t, \mathcal{F}_j)$. We refer the reader to the original paper [14] for details on the CMA-ES update algorithm. The best performing fitness score across all generations is chosen as the optimal pixel perturbations, δ^*. It must be noted that no samples from the testing data was used in the training phase for optimizing the noise generator parameters. During testing, we train with the optimally corrupted training data and perform inference on clean test data.

4. EXPERIMENTAL RESULTS

We evaluated our method on four datasets: MNIST, Fashion-MNIST, SVHN cropped 32×32 images, and CIFAR10 images. The perturbed MNIST images for $N_p = 1$ are shown in Figure 2 (a). Learning perturbations by evolution involves multiple training rounds in each generation. We used two custom CNN models as underlying models in the evolutionary learning stage: GrayNet (24C3-P-48C3-P-256FC-10S), for MNIST, Fashion-MNIST and ColorNet (32C3-32C3-P-64C3-64C3-P-128C3-128C3-P-512FC-10S) for CIFAR10, SVHN dataset. We use four settings of number of pixel perturbation, $N_p = \{1, 2, 5, 10\}$.

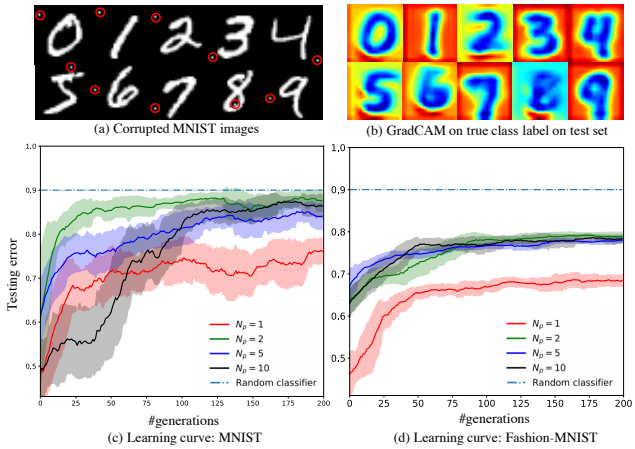

Fig. 2: (a) Highlighting learned single pixel perturbations on MNIST images, (b) GradCAM visualization of the last Conv layer for $N_p = 1$. Dominant gradient distribution is on the background. Learning curve with increasing generations of CMA-ES is shown for (c) MNIST and (d) Fashion-MNIST

Method	ResNet-20	ResNet-32	DenseNet-40
SVHN(clean)	93.5 ± 0.9	92.8 ± 1.0	92.3 ± 1.2
$N_p = 1$ [Baseline]	30.3 ± 8.6	41.4 ± 8.9	36.3 ± 4.2
$N_p = 1$ [7]	91.8 ± 0.2	90.9 ± 1.8	91.0 ± 0.4
$N_p = 1$, [ours]	31.3 ± 6.3	37.2 ± 10.4	32.1 ± 9.4
$N_p = 2$, [ours]	14.9 ± 2.4	18.4 ± 3.8	18.8 ± 4.7
$N_p = 5$, [ours]	$\mathbf{9.3 \pm 0.9}$	$\mathbf{11.0 \pm 0.3}$	$\mathbf{16.1 \pm 8.4}$

Table 1: Showing testing accuracy (in %) on clean test samples, trained on optimally perturbed samples with DA for 30 epochs on SVHN dataset. Experiments are repeated 3 times.

4.1. Learning Curves for Perturbation Optimization

We examine test error with increasing generations of our proposed algorithm as shown in Figure 2 (c) and Figure 2 (d) for MNIST and Fashion-MNIST datasets respectively. Test error is seen to grow as the evolutionary optimization advances indicating the soundness of our proposed optimization strategy. Additionally, we visualize the mean GradCAM distribution of 100 images per class from the testing dataset corresponding to the true class label for MNIST dataset in Figure 2 (b), which reveals that the CAM distribution shifts its density to non-salient background ROI in the image, thus learning non-discriminative features that do not generalize well. This might explain the drop in testing accuracy with increasing epochs.

4.2. Comparison to Prior Methods

As a baseline for our task, we choose a uniformly sampled spatial distribution of pixel perturbation, which is the starting point of the CMA-ES algorithm. Our method consistently out-

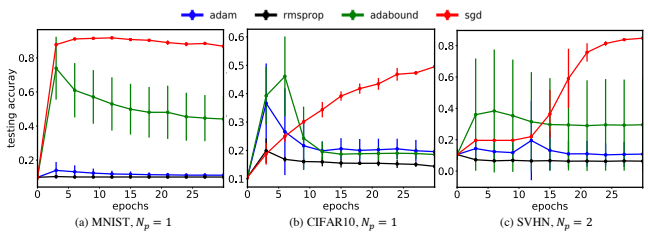

Fig. 3: Testing accuracy using various optimization strategies under proposed perturbation shows SGD consistently performs better than adaptive optimization techniques. Each experiment was performed 5 times and one std dev. error is shown.

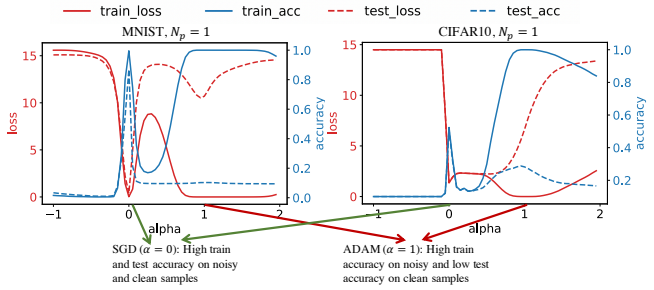

Fig. 4: Interpolating loss surface from SGD ($\alpha = 0$) to ADAM ($\alpha = 1$) weights. The loss surface around SGD parameter is sharper however has better generalization.

performs both the baseline method and Jacobsen et al. [7] on the metric of test error on the clean test set, for all the datasets as shown in Table 1. Our method shows superior performance compared to [7] because we perform optimization to search for the best corruption pattern whereas [7] uses heuristic pixel perturbations on the left-most column of the input image to encode class specific information. The baseline method outperforms Jacobsen et al. [7] due to data augmentation.

4.3. Adaptivity can Overfit to Training Perturbations

High out-of-sample error is generally attributed to poor convergence of the neural network parameters to an unfavorable local minimum. By examining the robustness of well-known optimization strategies to our proposed pixel perturbation algorithm, we wish to study if a certain algorithm is more liable to memorizing small perturbations while ignoring other salient statistical patterns in the training data. To this end, we trained CNN models on our proposed optimal ADS data using ADAM [19], SGD, RMSProp [20], and Adabound [21] optimization. The results are shown in Figure 3.

Wilson et al. [6] showed that adaptive methods are affected by spurious features that do not contribute to out-of-sample generalization by crafting a smart artificial linear regression example. Our method is an extension of such methods for automatic creation of spurious examples that scale to arbitrarily sized datasets by evolutionary strategies. Figure 3 reveals that ADAM and RMSProp show prohibitively low testing accuracy for all cases while vanilla SGD is surprisingly resilient to such perturbations showing better out-of-sample performance consistently for all the datasets. Adabound uses strategies from both SGD and Adam, thus showing intermediate performance. Thus, adaptive methods overfit to training perturbations while vanilla SGD is considerably robust to such changes.

Due to the input data corruption, the loss manifold changes to favor solutions that overfit to the spurious perturbation features. Our intuition is that adaptive methods adjust an algorithm to the geometry of the data [6] and thus overfits to such spurious features. In contrast, SGD's optimization strategy does not depend on the data, but it uses the l_2 geometry inherent to the parameter space. Thus it performs better than adaptive optimization algorithms.

Loss surface : Keskar et al. [3, 22] claimed that flatter minima solutions generalize better compared to its sharper counterparts. To investigate this phenomenon, we visualize the loss surface around the learned parameters by interpolating between the weights obtained from SGD and ADAM optimization following the strategy by Goodfellow et al. [23]. We plot the loss function values and train/test accuracies at intermediate intervals given as $\boldsymbol{w}_\alpha = \alpha \boldsymbol{w}_{\text{ADAM}} + (1-\alpha)\boldsymbol{w}_{\text{SGD}}$ as shown in Figure 4. Interestingly, we find that SGD finds sharper minima solutions where both test and train loss are low ($\alpha = 0$) compared to ADAM, where the train loss exhibits are more flatter geometry ($\alpha = 1$). This pattern is repeatedly visible for all datasets suggesting that sharpness of minima does not guarantee a solution that has better generalization robustness to training perturbations, which is along the same line of argument as claimed by Dinh et al. [4].

5. CONCLUSION

We present a population-based evolutionary strategy using a novel fitness score to search for pixel perturbations that explicitly maximize domain divergence and generalization gap. Our method incrementally fools the neural networks with each passing generation suggesting the existence of certain vulnerable spatial locations on input images. Our analysis reveals that a proper selection of neural network optimization is paramount to good generalization. We find that vanilla SGD performs significantly better than adaptive optimization methods in ignoring spurious training features that do not contribute to out-of-sample generalization. Our analysis of loss surface, reveals that in spite of good generalization performance SGD finds sharper minima solutions than ADAM. It might be tempting to conclude that sharper minima solutions are more robust to input perturbation overfitting however more analysis is required in this direction. We believe this work will fuel further research into understanding the generalization properties of deep learning optimization in the presence of input noise.

References

[1] Vladimir Vapnik, *The nature of statistical learning theory*, Springer science & business media, 2013.

[2] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[3] 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.

[4] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio, "Sharp minima can generalize for deep nets," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1019–1028.

[5] Nitish Shirish Keskar and Richard Socher, "Improving generalization performance by switching from adam to sgd," *arXiv preprint arXiv:1712.07628*, 2017.

[6] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht, "The marginal value of adaptive gradient methods in machine learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4148–4158.

[7] Jörn-Henrik Jacobsen, Jens Behrmann, Richard Zemel, and Matthias Bethge, "Excessive invariance causes adversarial vulnerability," *arXiv preprint arXiv:1811.00401*, 2018.

[8] Battista Biggio, Blaine Nelson, and Pavel Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[9] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 6103–6113.

[10] Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang, "Certified defenses for data poisoning attacks," in *Advances in neural information processing systems*, 2017, pp. 3517–3529.

[11] Thomas Tanay, Jerone TA Andrews, and Lewis D Griffin, "Built-in vulnerabilities to imperceptible adversarial perturbations," *arXiv preprint arXiv:1806.07409*, 2018.

[12] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 27–38.

[13] Pang Wei Koh and Percy Liang, "Understanding black-box predictions via influence functions," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1885–1894.

[14] Nikolaus Hansen, "The cma evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[15] David Ha and Jürgen Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems*, 2018, pp. 2450–2462.

[16] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira, "Analysis of representations for domain adaptation," in *Advances in neural information processing systems*, 2007, pp. 137–144.

[17] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[18] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan, "A theory of learning from different domains," *Machine learning*, vol. 79, no. 1-2, pp. 151–175, 2010.

[19] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] T Tieleman and G Hinton, "Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *Technical Report.*, 2017.

[21] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun, "Adaptive gradient methods with dynamic bound of learning rate," *arXiv preprint arXiv:1902.09843*, 2019.

[22] Sepp Hochreiter and Jürgen Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.

[23] Ian J Goodfellow, Oriol Vinyals, and Andrew M Saxe, "Qualitatively characterizing neural network optimization problems," *arXiv preprint arXiv:1412.6544*, 2014.