ADAPT: Adversarial Data Augmentation with Pseudo Labelling for Test-Time Adaptation

Anonymous CVPR submission

Paper ID *****

Abstract

This paper proposes adversarial data augmentations for test-time model adaptation using only unlabelled online target data. We progressively make the augmentations of the easy samples (low entropy with respect to the current model) harder using an adversarial setup and use these hard augmented views for self-training the model at test time. Our experiments on several benchmark datasets for classification and segmentation show the superiority of self-training with adversarial augmentations against the state-of-the art online test-time adaptation methods.

1. Introduction

Although deep neural networks (DNNs) give state-of-the-art results when tested on in-distribution data similar to the training data, their performance usually drops significantly after deployment on the unknown test domain with a sizeable co-variate shift. Several remedies such as data augmentation, regularization, and simulating distribution shifts during training have been proposed to make the model robust for test time deployment. However, simulating all possible distribution shifts that may occur after model deployment is very challenging and nearly impossible during model training on the source dataset. As a result, several recent works have instead focused on adapting the source-trained model to the unlabelled test domain. Among such approaches, online test-time adaptation (TTA) without access to the source dataset is the most practical and challenging. In this paper, we focus on this setting.

Test-Time Adaptation of a model is done by optimizing its parameters (or portion of parameters) using an unsupervised test-time objective function, e.g. entropy of prediction probabilities (TENT [3]), mutual information maximization (SHOT [2]), or cross-entropy using pseudo labels generated by a teacher model. However, as the model adapts to the target domain, its accuracy usually saturates and even starts declining due to over-fitting on the test-time objective. To overcome this issue, we propose to continually learn adversarial data augmentation policies by maximizing the test-time objective for the current adapted model on the target data. The augmented (hard) images are then used to self-train the model using the pseudo-labels derived from the momentum updated teacher model on the non-augmented (easy) images.

2. Related Work

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula

libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

3. Method

In this section, we describe our proposed methodology for learning adversarial test-time data augmentation followed by integrating it with our online test-time adaptation algorithm.

Problem Formulation Let $f = h \circ g$ denote the source model composed of an encoder $g : X \rightarrow Z$ that maps a given test image $x \in X$ to its feature representation $z \in Z$ and a classifier module $h : Z \rightarrow Y$ that maps z to its label $y \in Y$. Also, let θ_g and θ_h denote the parameters (all/partial) of the encoder (g) and classifier module (h) respectively that are updated during test-time adaptation by minimizing a given test-time objective $\mathcal{L}_{\text{Test}}$ as follows:

$$[\theta_g^{t+1}, \theta_h^{t+1}] \leftarrow [\theta_g^t, \theta_h^t] - \eta \cdot \nabla_{[\theta_g^t, \theta_h^t]} \mathcal{L}_{\text{Test}}(h \circ g(x)) \quad (1)$$

where η is the learning rate.

3.1. Learnable Test-Time Data Augmentation

Here we describe the augmentation policy search space comprising several image operations followed by our differentiable strategy to optimize and sample augmentations using an unsupervised test-time augmentation loss.

3.1.1 Policy Search Space

Let $\mathbb{O}$ be a set of image transformation operations $O : X \rightarrow X$ (e.g. Shear, Rotate, Brightness, Contrast, etc.) defined on the image space X. The magnitude of an image operation is denoted by $m \in [0, 1]$. Note that the output of some image operations may not depend on its magnitude (e.g. Equalize). We define a sub-policy ρ as a combination of N image transformation operations from $\mathbb{O}$ that is applied sequentially to a given image x as:

$$\rho(x, m^\rho) = O_1^\rho \dots O_N^\rho(x, m_N^\rho) \quad (2)$$

where $m^\rho = [m_1^\rho, \dots, m_N^\rho]$ denotes the magnitude of the sub-policy ρ. We define $\mathbb{P} = \{\rho_1, \rho_2, \dots\}$ as the set of all possible sub-policies. During test time adaptation, we sample k different sub-policies from $\mathbb{P}$ using a learnable multinomial distribution.

3.1.2 Policy Evaluation

Given a model f and a sup-policy ρ with magnitude m, a policy for an image x is evaluated by the following adversarial objective:

$$\mathcal{L}_{\text{Aug}}(x, \rho) = -H(\hat{y}, f(\rho(x, m))) + r(\rho(x, m), x) \quad (3)$$

where H denotes cross-entropy, $\hat{y}$ is the pseudo label and r is a regularization term for constraining augmentation severity. We use the distance between statistics of the model's L internal layers' activations of the augmented and non-augmented versions of image x.

$$r(\tilde{x}, x) = \frac{1}{L} \sum_{l=1}^{L} \|\mu_l(\tilde{x}) - \mu_l(x)\|^2 + \|\sigma_l(\tilde{x}) - \sigma_l(x)\|^2 \quad (4)$$

where $\tilde{x}$ is the augmented image, and μ_l and σ_l denotes the mean and standard deviation of the activations of l-th layer.

3.1.3 Policy Optimization

Let $M = [m^{\rho_1}, m^{\rho_1}, \dots, m^{\rho_{\|\mathbb{P}\|}}]$ denote magnitudes m^ρ of all sub-policies $\rho \in \mathbb{P}$ and $P = [p_1, p_2, \dots, p_{\|\mathbb{P}\|}]$ denote the probability of selecting a sub-policy. The expected policy evaluation loss (Eq. 3) for an image x over the policy search space is given by:

$$\mathrm{E}[\mathcal{L}_{\text{Aug}}(x)] = \sum_{i=1}^{\|\mathbb{P}\|} p_i \cdot \mathcal{L}_{\text{Aug}}(x, \rho_i) \quad (5)$$

Evaluating the gradient of $\mathrm{E}[\mathcal{L}_{\text{Aug}}(x)]$ w.r.t. M and P can become computationally expensive as $\|\mathbb{P}\| \sim \binom{\|\mathbb{O}\|}{N}$. Thus, we use the following reparameterization trick to estimate the unbiased gradient:

$$\begin{aligned} \nabla \mathrm{E}[\mathcal{L}_{\text{Aug}}(x)] &= \delta(x, \mathbb{P}) = \sum_{i=1}^{\|\mathbb{P}\|} \nabla(p_i \cdot \mathcal{L}_{\text{Aug}}(x, \rho_i)) \\ &= \sum_{i=1}^{\|\mathbb{P}\|} p_i (\nabla \mathcal{L}_{\text{Aug}}(x, \rho_i) + \mathcal{L}_{\text{Aug}}(x, \rho_i) \cdot \nabla \log p_i) \\ \Rightarrow \hat{\delta}(x, \rho_i) &= \nabla \mathcal{L}_{\text{Aug}}(x, \rho_i) + \mathcal{L}_{\text{Aug}}(x, \rho_i) \cdot \nabla \log p_i \quad (6) \end{aligned}$$

We randomly sample a policy ρ_i from $\mathbb{P}$ using probability distribution P and update parameters P and M using the following stochastic gradient rule:

$$[P, M] \leftarrow [P, M] - \gamma \cdot \hat{\delta}(x, \rho_i) \quad (7)$$

where γ is the learning rate.

3.2. Test Time Adaptation

4. Experiments

We empirically evaluate our model on three levels of domain shifts for two visual recognition tasks including image classification and segmentation. The levels of domain shifts comprise natural domain shifts, common image corruptions, and sym-to-real transfer.

4.1. Classification

Classification:

- Natural domain shits: Cifar10.1
- Common image corruptions: Cifar10-C, Cifar100-C, ImageNet-C
- sim-to-real: VisDA-C

Following the experiments of [1], we evaluate the model independence along the axis of training procedure and model architecture:

- training procedure: Normal, SimCLR, AugMix/AugMax
- model architecture: ResNet50, Vit-B, Efficient-Net, Wide-Resnet, ResNet18

4.2. Segmentation

Segmentation:

- Natural domain shifts: -
- Common image corruptions: Segmentation-C
- sim-to-real: VisDA-S

5. Ablation Study

We evaluate the effect of each component of the proposed method on the test-time performance.

- Optimal Hard Augmentations: We compare with (i) no augmentation, (ii) Randaugment policy, (iii) AutoAugment policy and (iv) GPS.
- Prototypes: we compare with no prototypes
- batch size: we campare with various batch size and various setup of batch-queue decoupling.
- sub-policy-dim: we evaluate our method with policy dimension from 1 to 5.
- aug multiplicity: we evaluate our method with 1 to 5 augmentation per image

6. Conclusion

TODO

Table 1. Classification accuracy (mean±std %) on VisDA-C train → val. Results are reported over three seeds. All methods use ResNet-50 backbone. For all experiments, parameters are updated with Stochastic Gradient Descent (SGD) with 0.9 momentum, 0.00025 learning rate and 128 batch size. Avg. denotes the average per-class top-1 accuracy.
* Tent++ denotes the Tent method with the usage of prototypes and class marginal maximization. It will be good to report std, number of seeds.

Method	plane	bicycle	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Avg.
(Online)													
Source	60.0±1.3	29.3±1.7	56.7±2.1	74.7±1.2	57.1±3.1	1.9±0.3	87.6±0.7	9.7±1.6	63.1±1.2	12.0±2.1	84.1±0.7	1.6±0.0	44.8±0.6
BN	84.0±0.2	58.7±0.6	74.7±0.1	51.4±0.1	82.3±0.4	50.9±0.2	85.2±0.1	52.3±0.1	70.2±0.2	44.0±0.5	80.2±0.1	23.0±0.3	63.1±0.1
Tent	84.4±0.3	59.5±0.4	76.1±0.1	52.5±0.1	83.0±0.2	53.9±0.1	86.7±0.0	55.2±0.3	73.4±0.2	45.3±0.4	80.4±0.3	21.8±0.2	64.3±0.1
SHOT	90.2±0.2	79.3±0.4	72.6±0.2	43.2±0.1	87.0±0.1	64.8±0.4	72.8±0.3	73.0±0.5	84.6±0.3	48.6±0.5	80.4±0.3	44.6±0.4	70.1±0.2
AdaContrast	94.7±0.2	77.9±0.5	81.4±0.2	66.2±0.3	94.2±0.2	85.2±0.6	87.7±0.2	75.6±0.8	90.6±0.2	41.3±0.3	85.6±0.0	26.1±0.2	75.5±0.1
Ours	94.3±0.1	82.8±1.2	78.4±1.0	61.0±0.7	91.2±0.3	90.3±0.6	81.6±1.2	79.7±0.6	91.8±0.5	52.7±0.6	83.5±0.5	32.9±1.2	76.7±0.1
Ours++	94.6±0.1	84.2±0.7	77.8±0.8	57.6±0.4	92.1±0.2	90.2±0.4	77.4±1.0	80.1±0.3	92.3±0.5	51.7±1.3	84.0±0.7	41.4±0.8	77.0±0.2

Table 2. Benefit comparison of our test-time augmentation module on pseudo-label based test time adaptation methods including SHOT [REF], AdaContrast [REF] and our proposed approach. Per-class average classification accuracy (mean±std %) on VisDA-C train → val is reported over 3 seeds. All methods use ResNet-50 backbone. VA denotes Vanilla Augmentation including Color Jittering, Random Horizontal Flip, and Random Resize Crop. RA denotes RandAugment [REF] and AA is AutoAugment [REF].

TTMA Algorithm	TTA Module			
	VA	RA	AA	Ours
SHOT	70.1±0.2	72.8±0.1	72.8±0.1	73.8±0.2
AdaContrast	75.5±0.1	75.3±0.2	75.6±0.3	75.8±0.2
Ours	75.3±0.1	75.8±0.2	75.9±0.1	76.7±0.1

Table 3. Ablation Study on VisDA-C train → val. Per-class average classification accuracy (mean±std %) is reported over 3 seeds. All methods use ResNet-50 backbone.

Cosine InfoMax	Pseudo Labelling	PL Refine	Denoise	Optimal TTA	Avg.
-	-	-	-	-	44.8±0.6
✓	-	-	-	-	69.7±0.1
✓	✓	-	-	-	72.6±0.1
✓	✓	✓	-	-	75.1±0.1
✓	✓	✓	✓	-	75.3±0.1
✓	✓	✓	✓	✓	76.7±0.1

References

[1] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8353, 2022. 3

[2] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 1

[3] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. 1