This ECCV 2018 paper, provided here by the Computer Vision Foundation, is the author-created version.
The content of this paper is identical to the content of the officially published ECCV 2018
LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

Domain transfer through deep activation matching

Haoshuo Huang[1][0000−0003−3945−3632], Qixing Huang[2][0000−0001−6365−8051], and Philipp Krähenbühl[2][0000−0002−9846−4369]

[1] Tsinghua University, Beijing, China
hhs14@mails.tsinghua.edu.cn
[2] University of Texas at Austin, Austin, USA
{huangqx,philkr}@cs.utexas.edu

Abstract. We introduce a layer-wise unsupervised domain adaptation approach for semantic segmentation. Instead of merely matching the output distributions of the source and target domains, our approach aligns the distributions of activations of intermediate layers. This scheme exhibits two key advantages. First, matching across intermediate layers introduces more constraints for training the network in the target domain, making the optimization problem better conditioned. Second, the matched activations at each layer provide similar inputs to the next layer for both training and adaptation, and thus alleviate covariate shift. We use a Generative Adversarial Network (or GAN) to align activation distributions. Experimental results show that our approach achieves state-of-the-art results on a variety of popular domain adaptation tasks, including (1) from GTA to Cityscapes for semantic segmentation, (2) from SYNTHIA to Cityscapes for semantic segmentation, and (3) adaptations on USPS and MNIST for image classification. [3]

Keywords: Domain adaptation, image classification, semantic segmentation, activation matching, GTA, SYNTHIA, Cityscapes, USPS and MNIST

1 Introduction

In this paper, we propose a novel approach for unsupervised domain adaptation. Our goal is to transfer a pre-trained network from a source domain, with an abundance of labels, to a relevant, but unlabeled target domain. This problem is inherently ill-posed, and the success or failure of domain adaptation is largely driven by assumptions placed on the source and target domains. A widely used assumption is that the underlying label distributions (e.g., from the output layer) of the source and target domains are similar (c.f. [1]). However, this assumption only provides a weak training signal for the target network. Because of this, existing techniques usually utilize additional generic constraints on network weights to make the training procedure better conditioned.

[3] The website of this paper is https://rsents.github.io/dam.html

The key idea of our approach is to align the activation distributions of intermediate layers. This strategy places more constraints on the target network, and thus improves the quality of the transferred network. Specifically, our approach aligns layer-wise distributions in two ways. First, we derive a closed-form matching criterion, under the assumption that the activation distribution is i.i.d. Gaussian. Second, we relax the i.i.d. Gaussian assumption, and match the em-

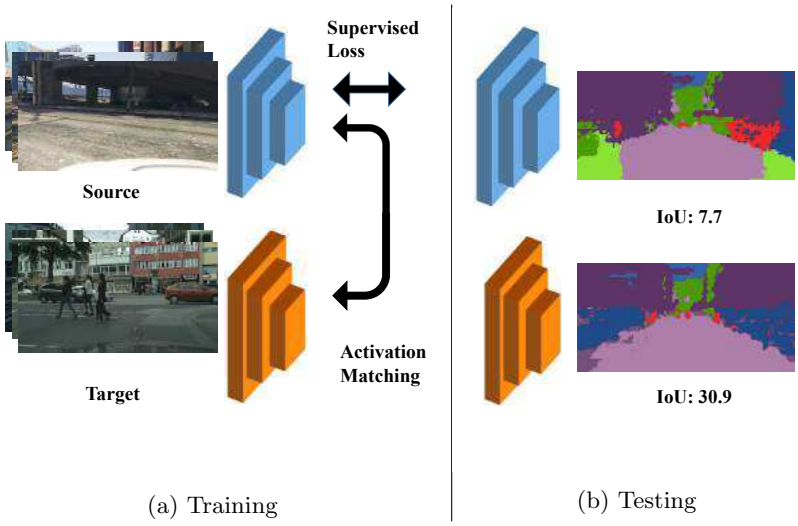


Fig. 1: Given a pretrained CNN in the source domain, we seek to adapt it to a target domain. In this example, the source domain consists of screenshots from the GTA game, while the target domain are real-word images from the Cityscapes dataset.

pirical distributions of activations using a Generative Adversarial Network [2]. Aligning activation distribution by itself is not enough, as distribution alignments only place modest constraints on network weights. There are multiple possible target networks that match activation distributions, many of which do not transfer any knowledge. This motivates us to include an additional regularizer on the target network, which keeps it close to the source throughout the training.

We evaluated the proposed approach on the tasks of image classification and dense semantic segmentation. In both cases, our approach out-performs state-of-the-art techniques for unsupervised domain adaptation. We also did an extensive ablation study, which demonstrates the importance of all components of our approach. Specifically, we show that matching intermediate activations always leads to a higher performance. Although regularization does not seems

to have significant effect in image classification, it improves both robustness and performance in semantic image segmentation.

2 Related Works

Transfer learning is a fundamental problem in machine learning with a wide range of applications in computer vision. It is beyond the scope of this paper to review all relevant works. We refer to [1, 3–12] for some recent advances and to [13, 14] for surveys on visual domain adaptation. In the following, we review recent works that are relevant to the approach presented in this paper.

Distribution-alignment based methods seek to align the source and target distributions in some common space, which provide regularization for training the target network. Thus, we can classify a distribution-based method based on the common spaces as well as the methods being used for aligning distributions. Saenko et al. [15] proposed a pairwise metric transform for visual domain adaptation. Early deep adaptive works align first and second order statistics for domain adaptation [16, 17]. More recent methods utilize generative adversarial networks [18] to align the source and target distributions [19]. Other distribution alignment methods include optimizing symmetric confusion metric [20] and the inverted label objective [1] . Our method differs from these methods in that we perform alignments across multiple layers in a deep network.

Map-based methods. Another solution to address unsupervised domain adaptation is to explicitly establish a map that aligns space of images in the source domain to the space of images in the target domain. This map allows us to transfer the labels from the source domain to the target domain either explicitly or implicitly. As a consequence, it allows us to train the network from the target domain using the labels from the source domain. Liu and Tuzel [21] performed weight sharing using hand-encoded layers to training a pair of generative models between two relative domains. For the task of image segmentation and classification, their method requires that some instances from the target domain are labeled. Ghifary et al. [22] used an additional reconstruction object in the target domain to prioritize distribution alignment in the unsupervised domain adaptation setting.

Another line of research applies generative adversarial networks to explicitly convert target images into source images. These approaches include the ones that learn from paired data [23–25] as well as from unpaired data [9, 26–30]. State-of-the-art techniques [29, 30] usually train a pair of maps between the source and target domains and enforce the consistency between them. Hoffman *et al.* [31] recently showed that combining image translation with unsupervised domain adaptation greatly improves the final accuracy of the adapted model. In experiments, we borrow this idea and combine our domain adaptation with image translation.

Visual domain adaptation for semantic segmentation. In contrast of the large body of works on visual domain adaptation for classification, less works have focused on visual domain adaptation for semantic segmentation. Levinkov and Fritz [32] first studied this problem by updating trained initial models during testing time using a sequential Bayesian model. Their method works well across weather conditions on similar road layouts. Hoffman et al. [33] and Ros et al. [34] pre-train a large model from multiple sources and then fine-tune on a sparsely labeled target domain via distillation and additional generic constraints on label distributions. Recently, Chen et al. [35] and Zhang et al. [5] align label distributions and/or class specific distributions as well as object priors for semantic segmentation. In contrast, we look into aligning distributions of activations of intermediate layers.

3 Overview

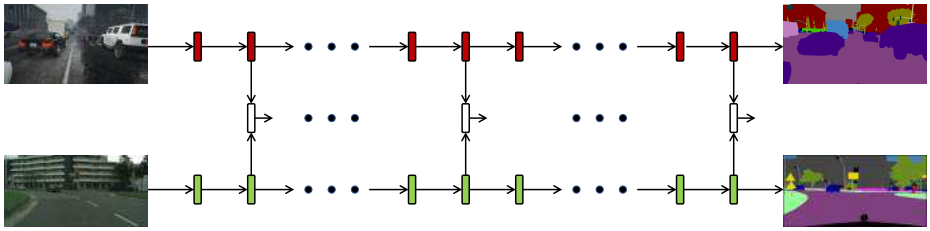


Fig. 2: Overview of the network architecture. (Top row) Network of the source domain. (Middle) Discriminators used to distinguish different dataset. (Bottom) Network of the target domain.

Consider a source domain X^s and a target domain X^t. With P^s and P^t we denote the empirical distributions of X^s and X^t, respectively. Instances from the source domain are labeled (e.g., with class labels or pixel-wise semantic labels). Suppose we have a task-specific network architecture F with L layers. Let F^s denote the pre-trained network architecture in the source domain, and θ^s its weights. To ease the discussion, we assume F^s is a feed-forward network, where f_i^s denotes the i-th layer. Extensions to DAG structured networks are straight forward. Let

$$F_i^s = f_i^s \circ \cdots \circ f_1^s.$$

be the sub-network that consists of the first i layers. Our goal is to learn a network F^t with parameters θ^t on the target domain. Since we do not have any label in the target domain, this problem is ill-posed. We thus constrain the problem in three ways.

Label distributions. We assume the underlying label distributions of the source and target domains are similar (e.g., distributions of class labels per-pixel), as is common for unsupervised domain adaptation [1]. This assumption clearly places certain constraints on the network of the target domain. However, it is easy to configure networks to match pixel-wise label distributions, without assigning any meaningful labels in the target domain. By itself, aligning label distributions is clearly insufficient.

Activation distributions. One of the key observations in this paper is that the domain shift does not just happen at the output layer, but anywhere inside the network. We address this by placing the additional constraint that the distributions of intermediate activations are similar between the source domain and the target domain. Such assumptions have been used [7] for specific layers and with simplified distributional assumptions. We propose to enforce it across all the layers for general activation patterns. While activation matching provides considerably more constraints on supervised domain adaption than merely aligning the label distributions, it is not yet sufficient — one can still design a target network so that it matches activation distributions but outputs different pixel-wise labels.

Weight drift The fundamental underlying assumption of domain adaptation is that the source representation carries some information about the target domain, and only needs to adapt slightly to perform well on the target. However, none of the above losses capture this gradual change. We thus add a regularizer between the source and target networks, to ensure that the filters do not change much during adaptation.

4 Approach

In this section, we present our approach for layer-wise unsupervised domain adaptation. We first present a general formulation. We then describe an effective two-stage approach that yields an approximate solution. Let $A_i^s = F_i^s(x^s)$ be the activation at the i-th layer of network F^s on a source image $x^s \in P^s$, and A_i^t be the corresponding activation of the target network on a different target image $x^t \in P^t$. Each spatial location is regarded as an i.i.d. sample. Let $P(A_i^s)$ and $P(A_i^t)$ be the distributions of these two activations over our entire source and target sets, respectively. Our objective is to match these distributions as well as possible, while keeping source and target networks close to each other. We express this in a constrained optimization framework:

$$\begin{aligned} \underset{\theta_t}{\text{minimize}} \quad & \|\theta^s - \theta^t\|^2 \\ \text{subject to} \quad & P(A_i^s) \approx P(A_i^t), \qquad 1 \leq i \leq L. \end{aligned} \tag{1}$$

We use $\approx$ to denote that the two distributions should match. This optimization problem is clearly hard. The major challenge lies on estimating high dimensional distributions of activation maps and matching them. We present two relaxations to this optimization problem.

4.1 Gaussian i.i.d. matching

A common practice for weight initialization of a neural network is that all activations are Gaussian i.i.d. [36]. More precisely, denote $A^s_{i,k}$ as the activation of channel k of layer i of the source network. , we assume $A^s_{i,k}$ follows a Gaussian distribution with mean $\mu^s_{i,k}$ and standard deviation $\sigma^s_{i,k}$. We denote $A^t_{i,k}$ as the corresponding activation from the target domain under the same Gaussian i.i.d. assumption. In this setting, matching these activation distributions simplifies to matching the mean and standard deviations of activations between the source and target domains. We do this by scaling and shifting activations.

Specifically, consider scaling the weights of the target network at layer i, F^t_i by a factor $\alpha_{i,k}$ and adding a bias $\beta_{i,k}$ for channel k. Each new target activation $\hat{A}^t_{i,k}$ is simply a shifted and scaled version of the old one:

$$A^t_{i,k} = \alpha_{i,k} A^t_{i,k} + \beta_{i,k}.$$

The same applies to the mean and variance:

$$\hat{\mu}^t_{i,k} = \alpha_{i,k}\mu^t_{i,k} + \beta_{i,k}, \quad \text{and} \quad \hat{\sigma}^t_{i,k} = \alpha_{i,k}\sigma^t_{i,k}.$$

This gives us a clear path to match source and target distributions with

$$\alpha_{i,k} = \frac{\sigma^s_{i,k}}{\sigma^t_{i,k}} \quad \text{, and} \quad \beta_{i,k} = \mu^s_{i,j} - \alpha_k \mu^t_{i,k}.$$

Under a Gaussian i.i.d. assumption, it is sufficient to shift and scale the output of each layer using the following transformation

$$\hat{F}^t_i(x^t) = \frac{\sigma^s_{i,k}}{\sigma^t_{i,k}} \left(F^t_i(x^t) - \mu^t_i\right) + \mu^s_i, \tag{2}$$

where μ_i and σ_i are the channel-wise mean and standard deviation in the source and target domain, respectively. Moreover, if the filters and activations are full rank, Equation (2) is a unique solution to Objective (1). Equation (2) reduces to AdaBN [37] and or more generally AutoDIAL [7] if applied directly to a batch normalization layer.

A major drawback of this simple matching is that the i.i.d. assumption ignores any structure in the data. Next, we show how to match the activation distributions in a more structured way.

4.2 General matching

Instead of directly matching the activations in Equation (1) using a hard constraint, we relax the constraint by minimizing a loss function. In this paper, we employ the Jensen-Shannon divergence (JSD) $J(P(A^s_i), P(A^t_i))$ for comparing two distributions. A nice property of JSD is that it is zero if and only if the two distributions match, and positive in all other cases. The new objective is

$$\min_{\theta^t} \|\theta^s - \theta^t\|^2 + \lambda \sum_{i=1}^{L} J(P(A^s_i), P(A^t_i)), \tag{3}$$

where λ measures the strength with which we enforce the constraint.

We optimize objective (3) using a generative adversarial networks (GAN) [2] The GAN effectively minimizes the Jensen-Shannon divergence in (3). It formulates a two player game between a generator, in our case the domain adaptation algorithm, and a discriminator that separates the source and target domains. GANs can be hard to train, particularly when the source and target distributions are different. We found that a careful initialization using the Gaussian activation matching, Section 4.1, was crucial for successful transfer. In addition, we used Least Square Generative Adversarial Network [38] for semantic segmentation, as it further stablized the training. For digit classification, a classical GAN [2] was sufficient.

5 Experimentals

We evaluate our approach on several tasks: digit image classification and semantic segmentation. For digit image classification we transfer among three datasets: MNIST [39], USPS [40] and SVHN [40]. These three datasets share a common label space corresponding to digits 0 to 9. MNIST and USPS feature grayscale handwritten digits, while SVHN contains color images of house numbers from Google Street View. We follow the evaluation protocol of ADDA [1] and transfer MNIST $\rightarrow$ USPS, USPS $\rightarrow$ MNIST and SVHN $\rightarrow$ MNIST. For each pair of datasets we report the classification accuracy on the target set.

For semantic segmentation we transfer between three datasets: Cityscapes [41], GTA [42] and SYNTHIA [43]. Cityscapes features real world scenes of a car driving through 50 European cities. GTA and SYNTHIA try to mimic Cityscapes as well as possible in simulation. While Cityscapes only contains 2975 pixel accurate training images, both synthetic datasets are considerably larger with 9400 for SYNTHIA and 24966 for GTA. We transfer semantic segmentation models from both synthetic datasets to Cityscapes. We evaluate our representation on the 1525 test images of Cityscapes, using three standard metrics: Intersection over Union (IoU) over the entire dataset, pixel-wise classification accuracy, and class-weighted classification accuracy.

For each task we compare to several baselines, and the prior state of the art. Our baselines include: The source model without adaptation, and a fully supervised model trained on the target domain. We compare to ADDA on all tasks, and the current state of the art [5] on Cityscapes. In addition, we also compare to a recent public available method CYCADA [31].

Network Architecture. For image classification, we adapt LeNet [39]. The choice of the discriminator is critical here. We adapt the same settings as ADDA [1], with a three-layer network, i.e., two 500-unit hidden layers and one final classification layer. We only add the discriminators in the last two layers of the source and target networks. In order to balance the influence of different activation maps, we multiply 0.1 to the scale of penultimate layer. For classification we pre-train the discriminator for 500 iterations, before training it jointly with the

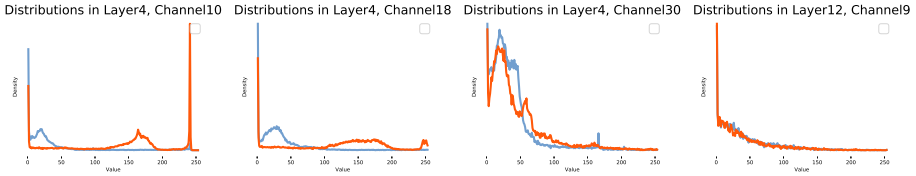


Fig. 3: Can you tell if two images are from the same domain or not by just looking at a single activation inside the network? Here, we show two histograms (orange or blue) of activations for a specific unit in the network. The histograms either come from two different images in the same domain, or different domains. Can you tell which is which? See footnote[4]for the result.

target network. Adam optimizer was used for training. For digit classification, we found it to be more stable to add one GAN at a time, after the training for the previous GAN converged.

We use ERFNet [42] for semantic segmentation. ERFNet consists of multiple down-sampling layers and residual-like modules. Compared to other networks for semantic segmentation, ERFNet provides a desired balance between segmentation accuracy and efficiency. For computational reasons, we skip the decoder of the ERFNet and use a simple bi-linear up-sampling. We choose Least Square Generative Adversarial Network [38] as our discriminator. GPU memory is the main limitation on the number of discriminators. We evenly distribute the discriminators among all layers for the segmentation task(4th/17, 8th/17, 12th/17, 17th/17) and it worked well enough. We explored various locations of the discriminator for digit classification, and the last two layers worked best.

Hyper-parameters We exhaustively explored the hyper-parameter λ, and found $\lambda = 0.1$ with a batch of 12 images yielded the best result. The method is quite robust, with all of these settings coming within 3% of the optimal setting.

We start the evaluation by verifying the core premise of this paper: A Domain Shift occurs throughout all layers of a network, not just the final layer. To test this premise we devise a little game called: Source-or-Not.

5.1 Source-or-Not

To illustrate that intermediate activations mismatch throughout a network between source and target domain we devised a little game: Source-or-Not. For a network pre-trained on a source domain X^s, we pick a random layer l, and a random unit i within that layer. We then choose two images either from the same or different domains, and plot the distribution of activations $A_{l,i}$ of that single unit across all spatial locations. The objective of the game is to tell from the activation distribution, if the two images came from the same dataset or from different ones. Figure 3 shows an example of this. After some calibration, a human observer almost exclusively wins the Source-or-Not game.

Approach	MNIST→USPS	USPS→MNIST	SVHN→MNIST
Supervised baseline	96.4	99.9	99.9
No adaptation	77.8	70.7	60.3
ADDA [1]	90.2 ± 0.9	97.2 ± 0.4	72.0 ± 0.6
ADDA [1] + **our regularization**	90.5 ± 0.5	97.4 ± 0.6	73.5 ± 0.8
Long et al. [44]	85.0	90.9	69.9
GAM (no regularization)	95.6 ± 0.6	97.8 ± 0.6	73.6 ± 0.6
GAM (full)	$\mathbf{95.7} \pm 0.5$	$\mathbf{98.0} \pm 0.5$	$\mathbf{74.6} \pm 1.1$

Table 1: Classification accuracy in percentage for transfer between MNIST, USPS, and SVHN. Higher is better. GAM stands general activation matching.

We will look how our deep activation matching deals with the domain shift. We start with digit classification experiments.

5.2 Classification

For digit classification, we compare our methods, Gaussian i.i.d. matching and General Activation Matching (or *GAM*), to the current state of the art, ADDA [1], the algorithm of Long et al. [44] and various ablations of our algorithm. For ADDA, we report the performance of running their code on our platform. However, we got slightly different results than reported in the original paper, despite running their code as is.

We use almost the same settings as ADDA. We use batch size 128 and a learning rate of 1e4. SGD is used with a momentum of 0.9. The model is trained for 20000 iterations in all tasks. We scaled 0.1 for GAN losses for the next-to-last layer. We half the weight regularization losses for all layers.

Table 1 shows our results. Compared to results reported by ADDA, our baseline performs significantly better for USPS to MNIST, while for SVHN to MNIST it does slightly worse. For each task, domain adaptation leads to a significant boost over a source-only model, bridging the gap between source- and target- trained models by over two thirds. Adding our weight regularization term to the baseline already gives ADDA a slight boost. However the most significant boost throughout all domains comes from the general activation matching using adversarial networks.

This clearly establishes that matching the distribution of intermediate layers in a deep network matters for domain adaptation.

Distributional mismatch An underlying assumption in our work is that the source and target labels follow the same distribution. This is true for the nicely balanced MNIST, SVHN, and USPS, but might not hold in general. To study

[4] Solution of the Source-or-Not game in Figure 3: First 2 - cityscapes vs GTA, last 2 - Cityscapes vs Cityscapes.

Source	Target	no adaptation	**GAM** (full)
Odd SVHN	Full MNIST	32.5	**36.9**
Even SVHN	Full MNIST	34.1	**37.6**
Full SVHN	Odd MNIST	**50.8**	29.8
Full SVHN	Even MNIST	**69.8**	44.0
Odd MNIST	Full USPS	47.8	**50.5**
Even MNIST	Full USPS	37.1	**42.8**
Full MNIST	Odd USPS	**87.6**	72.8
Full MNIST	Even USPS	73.7	**80.1**

Table 2: Domain adaptation with mismatched label distributions. Full uses the original dataset, odd removes half of the odd digits, even removes half of the even digits.

the effect of a distributional mismatch between training and testing we trained and evaluated on subsets of the datasets with skewed label distributions. We tried three subsets for each dataset: The full dataset (Full), half of the odd digits removed (Odd), half of the even digits removed (Even). The test set was unchanged. The results are summarized on Table 2.

Training on anything other than the full dataset significantly drops the generalization performance of both source-only classifier and our adapted model. However, here adaptation is able to recover a significant part of the lost performance. This shows that our GAM does not overly rely on the distributions matching exactly, but is able to tolerate some distributional mismatch. However, when the target and test set do not match (Odd and Even Target) our method fails, as it adapts to the wrong test distribution.

Next, we show how GAM performs on semantic segmentation.

5.3 Semantic segmentation

Domain adaptation generally assumes that the source and target domains share similar label distributions. However this might not always be true. We first establish a baseline for the optimal (oracle) classifier that perfectly matches the source label distribution.

Oracle performance To compute this oracle performance we first count label frequencies in both the source and target domain. Let n_l^s and n_l^t be the number of pixels labeled l in the source and target domains, respectively. For each label l, the maximal intersection in the IoU score is $\max(n_l^s, n_l^t)$, while the union is $n_l^s + n_l^t - \max(n_l^s, n_l^t)$. This allows us to compute an upper bound on the IoU, pixel- and class-wise accuracy without labeling a single image. Table 3 shows the result. If we perfectly follow GTA label distribution we can never exceed 55% IoU accuracy, or 82% pixel-wise accuracy.

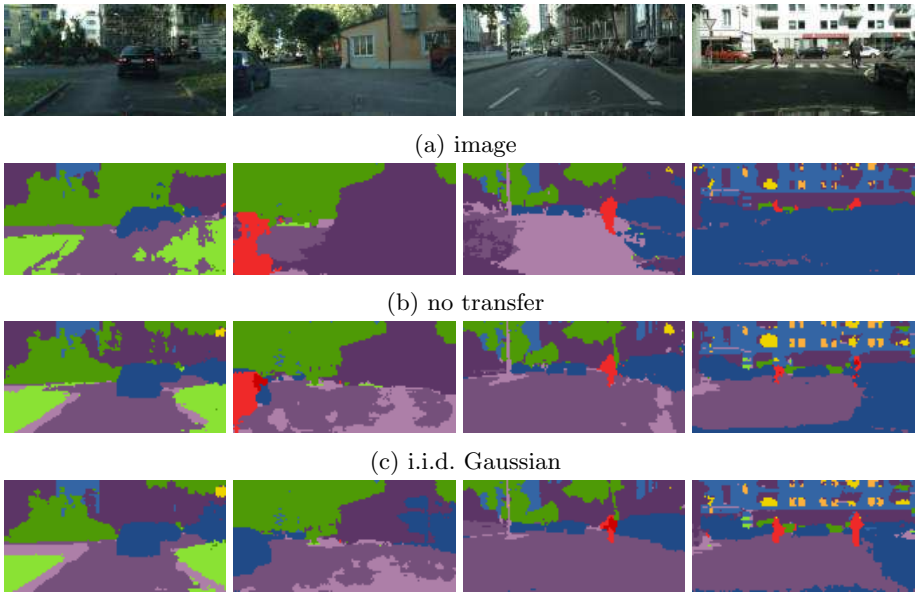
Experiments	Arch.	Global accuracy	Class accuracy	IoU
Oracle (source distribution)		82	72	55
Supervised baseline	A	-	-	65.0
No transfer	A	45	28.9	15.8
i.i.d Gaussian matching	A	72	41.3	28.0
ADDA	A	79.4	42.6	28.6
ADDA + **our regularization**	A	80.3	42.4	30.5
GAM (full)	A	**80.6**	**44.2**	**31.3**
Curriculum [5] (no transfer)	B	–	-	23.1
Curriculum [5]	B	–	-	28.9
No transfer (ours)	B	–	–	18.8
GAM (full)	B	80.9	43.8	**32.6**
CyCADA [31]	C	**82.3**	**72.4**	39.5
ADDA	C	–	–	39.2
GAM (full)	C	81.1	73.1	**40.2**

Table 3: Experimental results of different models evaluating in Cityscapes datasets. We fine-tune all the experiments from the original model, which only have 15.8 IoU. Different evaluation metrics are used in evaluation. Our approach achieves the best performance in every evaluation metric. Architecture A is ERFNet, B is VGG16-FCN8s, C is Dilated Residual Network.

Baselines Here, we again compare to a fully supervised baseline, trained using target labels, a baseline without any transfer, ADDA, and the current state-of-the-art, curriculum domain adaptation [5] and CyCADA [31].

GTA to Cityscapes We randomly cropped images to 1024×512 and feed them to networks. We use ADAM optimizer with a batch size of 12. As for this task, every discriminator has the same scale and we did not find better performance when we changed the scale. We also doubled the weight regularization loss for all layers.As shown in Table 3, the transfer from GTA to Cityscapes is much more challenging than digit classification. The baseline algorithm without any domain transfer results in a drop of 16 in terms of IoU accuracy, while a fully supervised model achieves 65.0. However the performance almost doubles through the simple i.i.d Gaussian matching. The Gaussian matching performs nearly as well as the best prior work, ADDA. Adding our regularizer to ADDA again boosts its performance, but not as much as our complete General Activation Matching. Here, the prior state of the art trained a slightly different baseline model, performing at 23.1% IoU without transfer, however their transfer algorithm does not lead to a large improvement on top of the initialization.

Both the ADDA baseline and our General Activation Matching do not perform well without a regularization term. This is in part due to filters collapsing as we train the adversarial network. Figure 5 shows how the regularization term helps both ADDA and our method train longer without a collapse in transfer



(a) image

(b) no transfer

(c) i.i.d. Gaussian

(d) General Activation Matching

Fig. 4: Qualitative results. Cars are blue, buildings are gray. Roads are purple and the sidewalks are dark purple, trees green. The quantitative improvement is directly reflected in the increased segmentation quality of our transferred model.

accuracy. For ADDA, relatively early in training, many of the filters and activations go towards zero and do not recover. With early stopping, the resulting model performs only marginally better than the no-adaptation baseline. For a fair comparison, we compare to the ADDA at peak performance, before filters collapsed.

Curriculum uses a slightly different architecture. We use their architecture to both train a source model from scratch and adapt the model using GAM. GAM significantly outperforms Curriculum despite a lower baseline (no transfer) performance.

Finally, we compare to CyCADA and ADDA on the Dilated Residual Network [45]. We follow Hoffman*et al.* [31] and pretrain our source model on translated images of the CycleGAN [29] model. We also provide an ADDA baseline in this training setup. GAM again outperforms all prior works and shows state of the art performance.

Figure 4 shows a visual comparison among different transfer algorithms. While the gains of transfer learning from GTA to Cityscapes are impressive, we are still far from the supervised performance. A reason for this might be that the source dataset is too different from the target. This motivates us to try another synthetic dataset.

Experiments	Global accuracy	Class accuracy	IoU
Source only	77.30	59.21	45.19
GAM (full)	**92.67**	**74.32**	**62.26** (+17.07)
Curriculum [5] (no transfer)	-	-	17.4
Curriculum [5]	-	-	29.0 (+11.6)
GAM (finetune from [5])	-	-	30.7

Table 4: Transfer from SYNTHIA to Cityscapes. We compare to the state of the art using a slightly different baseline model. Despite this our algorithm yields a larger improvement over the baseline.

SYNTHIA to Cityscapes For SYNTHIA we pre-train the model on the same 22 source classes as Zhang *et al.* [5], and transfer the same 16 classes to Cityscapes. This setup is slightly different from the GTA to Cityscapes experiment, where we transferred all classes.

We compare to Curriculum domain adaptation, the current state-of-the-art, using the same models as in previous experiments. Table 4 shows the results. In this setup, our baseline performs significantly better than the reported state of the art. In addition, we also achieved a larger absolute improvement from our domain adaptation algorithm. This is in part due to the poor performance of the baseline model in Curriculum. If we finetune GAM on that same baseline, we see only a modest increase in performance over the full Curriculum system.

We have clearly established that Deep Activation Matching performs at, or higher than the current state of the art in unsupervised domain adaptation. In a final experiment, we see how well our approach compares to fine-tuning on a small set of labeled target images.

5.4 Comparison to fine tuning

The final question we would like to address in this paper is: How many labeled images is a state-of-the-art transfer learning algorithm worth. The answer is $30-60$, as we show in Figures 6. Fine-tuning a pre-trained model on just $30-60$ images will do as well as transfer learning on hundreds. In other words, if the authors would have labeled Cityscapes images, instead of writing this paper they would have obtained a higher transfer learning performance.

However, this is only part of the story. First, as Figure 6 shows, fine-tuning our transferred representation, still yields a boost of $2-3\%$ in accuracy. Second, this experiment assumes that we have labels in the target domain, which is only true for proxy-tasks we study in computer vision, and might not hold for robotics or autonomous driving tasks.

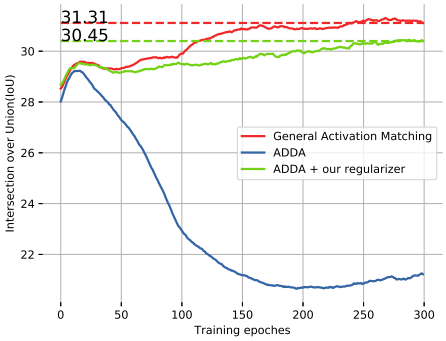


Fig. 5: Test accuracy of our model over several training iterations. Learning our General Activation matching without a regularization term leads to a collapse in filters, and diminishes performance.

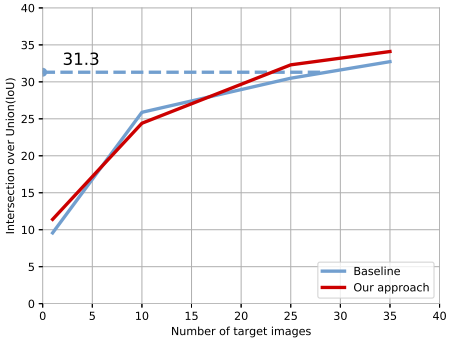


Fig. 6: Comparison of our state of the art domain transfer algorithm to fine-tuning on a limited set of target images. We fine-tune both our representation, and the baseline without any transfer. Our representation starts out below the transfer baseline due to overfitting.

6 Conclusions

In summary, we propose a novel approach for domain adaptation, based on closed form or adversarial activation matching of activation functions. Our experiments show that we can significantly outperform the state-of-the-art both in terms of robustness and performance.

There are ample opportunities for future research. For example, it would be interesting to study other ways to matching activation functions. In addition, which layers to match activation functions desire deeper analysis. Finally, so far we have studied domain adaptation among two networks, the same idea can be applied to match activation functions across multiple domains.

Acknowledgment

We would like to thank Angela Lin, and Thomas Crosley for their valuable comments and feedback on this paper. This work was supported in part by Berkeley DeepDrive and an equipment grant from Nvidia.

References

1. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. CVPR (2017)
2. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014)
3. Panareda Busto, P., Gall, J.: Open set domain adaptation. In: ICCV. (2017)
4. Gebru, T., Hoffman, J., Fei-Fei, L.: Fine-grained recognition in the wild: A multi-task domain adaptation approach. In: ICCV. (2017)
5. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. In: ICCV. (2017)
6. Gholami, B., (Oggi) Rudovic, O., Pavlovic, V.: Punda: Probabilistic unsupervised domain adaptation for knowledge transfer across visual categories. In: ICCV. (2017)
7. Maria Carlucci, F., Porzi, L., Caputo, B., Ricci, E., Rota Bulo, S.: Autodial: Automatic domain alignment layers. In: ICCV. (2017)
8. Yan, H., Ding, Y., Li, P., Wang, Q., Xu, Y., Zuo, W.: Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In: CVPR. (2017)
9. Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. CVPR (2017)
10. Herath, S., Harandi, M., Porikli, F.: Learning an invariant hilbert space for domain adaptation. In: CVPR. (2017)
11. Koniusz, P., Tas, Y., Porikli, F.: Domain adaptation by mixture of alignments of second- or higher-order scatter tensors. In: CVPR. (2017)
12. Sankaranarayanan, S., Balaji, Y., Castillo, C.D., Chellappa, R.: Generate to adapt: Aligning domains using generative adversarial networks. CVPR (2018)
13. Patel, V.M., Gopalan, R., Li, R., Chellappa, R.: Visual domain adaptation: A survey of recent advances. IEEE Signal Process. Mag. (2015)
14. Csurka, G.: Domain adaptation for visual applications: A comprehensive survey. arXiv preprint arXiv:1702.05374 (2017)
15. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. (2010)
16. Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., Darrell, T.: Deep domain confusion: Maximizing for domain invariance. CoRR (2014)
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR, IEEE Computer Society (2015)
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014)
19. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. JMLR (2016)
20. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. ICCV (2015)
21. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: NIPS. (2016)
22. Ghifary, M., Kleijn, W.B., Zhang, M., Balduzzi, D., Li, W.: Deep reconstruction-classification networks for unsupervised domain adaptation. In: ECCV. (2016)
23. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. CVPR (2017)

24. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. CVPR (2017)
25. Karacan, L., Akata, Z., Erdem, A., Erdem, E.: Learning to generate images of outdoor scenes from attributes and semantic layouts. CoRR (2016)
26. Yoo, D., Kim, N., Park, S., Paek, A.S., Kweon, I.S.: Pixel-level domain transfer. In: ECCV. (2016)
27. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation. ICLR (2017)
28. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. CVPR (2017)
29. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. ICCV (2017)
30. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for image-to-image translation. ICCV (2017)
31. Hoffman, J., Tzeng, E., Park, T., Zhu, J., Isola, P., Saenko, K., Efros, A.A., Darrell, T.: Cycada: Cycle-consistent adversarial domain adaptation. ICML (2018)
32. Levinkov, E., Fritz, M.: Sequential bayesian model update under structured scene prior for semantic road scenes labeling. In: ICCV. (2013)
33. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. CoRR **abs/1612.02649** (2016)
34. Ros, G., Stent, S., Alcantarilla, P.F., Watanabe, T.: Training constrained deconvolutional networks for road scene semantic segmentation. CoRR **abs/1604.01545** (2016)
35. Chen, Y.H., Chen, W.Y., Chen, Y.T., Tsai, B.C., Frank Wang, Y.C., Sun, M.: No more discrimination: Cross city adaptation of road scene segmenters. In: ICCV. (2017)
36. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AIStats. (2010)
37. Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. ICLR (2017)
38. Mao, X., Li, Q., Xie, H., Lau, R.Y.K., Wang, Z.: Multi-class generative adversarial networks with the L2 loss function. CoRR (2016)
39. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE (1998)
40. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Deep Learning Workshop. (2011)
41. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016)
42. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. CoRR (2016)
43. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.: The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR. (2016)
44. Long, M., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. ICML (2016)
45. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: CVPR. (2017)