A Closer Look at Smoothness in Domain Adversarial Training

Harsh Rangwani[*,1] Sumukh K Aithal[*,1,2] Mayank Mishra[1] Arihant Jain[1]
R. Venkatesh Babu[1]
[1]Video Analytics Lab, IISc, Bengaluru, India; [2]PES University, Bengaluru, India

Abstract

Domain adversarial training has been ubiquitous for achieving invariant representations and is used widely for various domain adaptation tasks. In recent times, methods converging to smooth optima have shown improved generalization for supervised learning tasks like classification. In this work, we analyze the effect of smoothness enhancing formulations on domain adversarial training, the objective of which is a combination of task loss (eg. classification, regression etc.) and adversarial terms. We find that converging to a smooth minima with respect to (w.r.t.) task loss stabilizes the domain adversarial training leading to better performance on target domain. In contrast to task loss, our analysis shows that converging to smooth minima w.r.t. adversarial loss leads to sub-optimal generalization on the target domain. Based on the analysis, we introduce the Smooth Domain Adversarial Training (SDAT) procedure, which effectively enhances the performance of existing domain adversarial methods for both classification and object detection tasks. Our analysis also provides insight into the extensive usage of SGD over Adam in the community for domain adversarial training. Code: https://github.com/val-iisc/SDAT

1. Introduction

Unsupervised Domain Adaptation refers to the class of methods that enables the model to learn representations from the source domain's labeled data that generalizes well on the unseen data from the target domain [1, 32–34, 55]. A prominent line of work is based on Domain adversarial training (DAT) [16].

DAT involves using an additional discriminator to distinguish between source and target domain features. A Gradient Reversal layer (GRL) is introduced to achieve the goal of learning domain invariant features. The follow-up works have improved upon this basic idea by introducing a class information-based discriminator (CDAN [34]), introducing a transferable normalization function [50], using an improved Margin Disparate Discrepancy [55] measure between source and target domain, etc.

As DAT objective is a combination of Generative Adversarial Network (GAN) [19] (adversarial loss) and Empirical Risk Minimization (ERM) [48] (task loss) objectives, there has not been much focus on explicitly analyzing and improving the nature of optimization in DAT. In optimization literature, it has been often stated that smoother minima generalizes better on unseen data [13,20,22,23]. Recently, a method called Sharpness Aware Minimization (SAM) [15] has been proposed for improved generalization, which finds smoother minima with an additional gradient computation step. However, we find that naively using smoothness techniques (like SAM) on DAT does not lead to improved generalization on target domain.

In this work, we analyse the loss landscape near the optimal point obtained by DAT by inspecting the eigenspectrum of Hessian (i.e. curvature) of the task loss (ERM term for classification). Based on the insights gained through our analysis, we summarise our contributions as:

- We show that converging to smooth minima w.r.t. task loss leads to stable and effective domain alignment through DAT, whereas smoothness enhancing formulation for adversarial loss leads to sub-optimal performance via DAT. We also find that using Stochastic Gradient Descent (SGD) as optimizer converges to a smoother minima in comparison to Adam [30].
- For enhancing the smoothness w.r.t. task loss near optima in DAT, we propose a simple, novel, and theoretically motivated SDAT (Fig. 1) formulation that leads to stable DAT resulting in improved generalization on the target domain.
- We find that SDAT, when combined with the existing state-of-the-art (SOTA) baseline for DAT, leads to significant gains in performance. Notably, with ViT backbone, SDAT leads to an average gain of **3.1%** over baseline, producing SOTA DA performance without requiring any additional module (or pre-training data) using only a 12 GB GPU. Moreover, we show a prototypical application of SDAT in DA for object detection, showing it's diverse applicability across various tasks.

[*] Equal Contribution, Correspondance to : harshr@iisc.ac.in

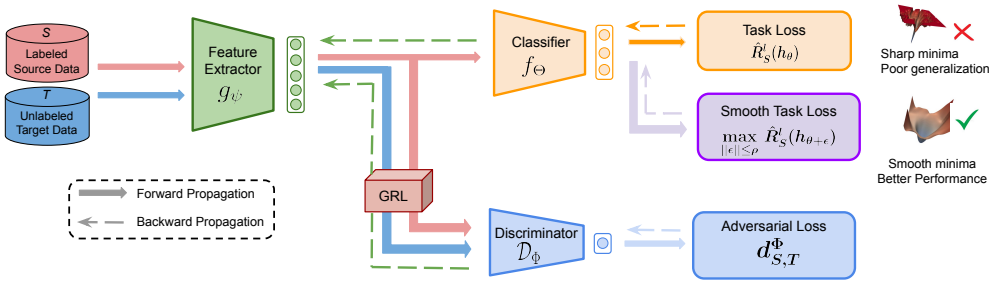


Figure 1. Overview of Smooth Domain Adversarial Training (SDAT). We demonstrate that converging to smooth minima w.r.t. adversarial loss leads to sub-optimal DAT. Due to this conventional approaches which smooth combination of task loss and adversarial loss lead to sub-optimal results. Hence, we propose SDAT which only focuses on smoothing task loss, leading to stable training which results in effective generalization on target domain. [1]

2. Background

2.1. Preliminaries

We will primarily focus on unsupervised DA where we have labeled source data $S = \{(x^s_i, y^s_i)\}$ and unlabeled target data $T = \{(x^t_i)\}$. The source samples are assumed to be sampled i.i.d. from source distribution P_S defined on input space $\mathcal{X}$, similarly target samples are sampled i.i.d. from P_T. $\mathcal{Y}$ is used for denoting the label set which is $\{1, 2, \ldots, k\}$ in our case as we perform multiclass (k) classification. We denote $y : \mathcal{X} \to \mathcal{Y}$ a mapping from images to labels. Our task is to find a hypothesis function h_θ that has a low risk on the target distribution. The source risk (a.k.a expected error) of the hypothesis h_θ is defined with respect to loss function l as: $R^l_S(h_\theta) = \mathbb{E}_{x\sim P_S}[l(h_\theta(x), y(x))]$. The target risk $R^l_T(h_\theta)$ is defined analogously. The empirical versions of source and target risk will be denoted by $\hat{R}^l_S(h_\theta)$ and $\hat{R}^l_T(h_\theta)$. All notations used in paper are summarized in App. A. In this work we build on the DA theory of [1] which is a generalization of [2]. We first define the discrepancy between the two domains.

Definition 2.1 ($D^\phi_{h_\theta,\mathcal{H}}$ discrepancy)**.** *The discrepancy between two domains P_S and P_T is defined as following:*

$$D^\phi_{h_\theta,\mathcal{H}}(P_S||P_T) := \sup_{h'\in\mathcal{H}} [\mathbb{E}_{x\sim P_S}[l(h_\theta(x), h'(x))]] - [\mathbb{E}_{x\sim P_T}[\phi^*(l(h_\theta(x), h'(x)))]] \quad (1)$$

Here ϕ^* is a frenchel conjugate of a lower semi-continuous convex function ϕ that satisfies $\phi(1) = 0$, and $\mathcal{H}$ is the set of all possible hypothesis (i.e. Hypothesis Space).

This discrepancy distance $D^\phi_{h_\theta,\mathcal{H}}$ is based on variational formulation of f-divergence [37] for the convex function ϕ. The $D^\phi_{h_\theta,\mathcal{H}}$ is the lower bound estimate of the f-divergence function $D^\phi(P_S||P_T)$ (Lemma 4 in [1]). We state a generalization bound on target risk $R^l_T(h_\theta)$ based on proposed $\mathcal{D}^\phi_{h_\theta,\mathcal{H}}$ discrepancy [1] for it's soundness in App. C.

2.2. Unsupervised Domain Adaptation

In this section, we first define the components of the framework we use for our purpose: $h_\theta = f_\Theta \circ g_\psi$ where g_ψ is the feature extractor and f_Θ is the classifier. The domain discriminator $\mathcal{D}_\Phi$, used for estimating the discrepancy between P_S and P_T is a classifier whose goal is to distinguish between the features of two domains. For minimizing the target risk (Th. 1), the optimization problem is as follows:

$$\min_\theta \mathbb{E}_{x\sim P_S}[l(h_\theta(x), y(x))] + D^\phi_{h_\theta,\mathcal{H}}(P_S||P_T) \quad (2)$$

The discrepancy term under some assumptions (refer App. B) can be upper bounded by a tractable term:

$$D^\phi_{h_\theta,\mathcal{H}}(P_S||P_T) \leq \max_\Phi d^\Phi_{S,T} \quad (3)$$

where $d^\Phi_{S,T} = \mathbb{E}_{x\sim P_S}[\log(\mathcal{D}_\Phi(g_\psi(x)))] + \mathbb{E}_{x\sim P_T} \log[1 - \mathcal{D}_\Phi(g_\psi(x))]$. This leads to the final optimization of:

$$\min_\theta \max_\Phi \mathbb{E}_{x\sim P_S}[l(h_\theta(x), y(x))] + d^\Phi_{S,T} \quad (4)$$

The first term in practice is empirically approximated by using finite samples $\hat{R}^l_S(h_\theta)$ and used as task loss (classification) for minimization. The empirical estimate of the second term is adversarial loss which is optimized using GRL as it has a min-max form. (Overview in Fig. 1) The above procedure composes DAT, and we use CDAN [34] as our default DAT method.

3. Analysis of Smoothness

In this section, we analyze the curvature properties of the task loss with respect to the parameters (θ). Specifically, we focus on analyzing the Hessian of empirical source risk $H = \nabla^2_\theta \hat{R}^l_S(h_\theta)$ which is the Hessian of classification (task) loss term. For quantifying the smoothness, we measure the trace $Tr(H)$ and maximum eigenvalue of Hessian (λ_{max}) as a proxy for quantifying smoothness. This is motivated by analysis of which states that the low value of λ_{max} and $Tr(H)$ are indicative of convergence to highly smooth loss landscape [26]. Based on our observations we articulate our conjecture below:

Conjecture 1. *Low $\lambda_{\max}$ for Hessian of empirical source risk (i.e. task loss) $\nabla^2_\theta \hat{R}^l_S(h_\theta)$ leads to stable and effective DAT, resulting in reduced risk on target domain $\hat{R}^l_T(h_\theta)$.*

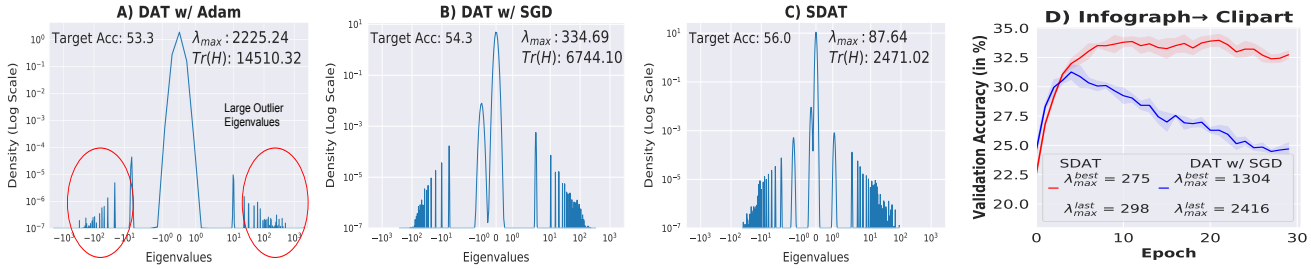


Figure 2. Eigen Spectral Density plots of Hessian ($\nabla^2 \hat{R}_S^l(h_\theta)$) for Adam (**A**), SGD (**B**) and SDAT (**C**) on Art → Clipart. Each plot contains the maximum eigenvalue (λ_{max}) and the trace of the Hessian ($Tr(H)$), which are indicators of the smoothness (Low $Tr(H)$ and λ_{max} indicate the presence of smoother loss surface). Low range of eigenvalues (x-axis), $Tr(H)$ and λ_{max} for SGD compared to Adam indicates that it reaches a smoother minima, which leads to a higher target accuracy. D) Validation accuracy and $\lambda_{\max}$ comparison for SDAT and DAT across epochs, SDAT shows significantly stable training with low $\lambda_{\max}$.

For empirical verification of our conjecture, we obtain the Eigen Spectral Density plot for the Hessian $\hat{R}_T^l(h_\theta)$. We show the λ_{max}, $Tr(H)$ and Eigen Spectrum for different algorithms, namely DAT w/ Adam, DAT w/ SGD and our proposed SDAT (Fig. 2). We find that *high smoothness leads to better generalization on the target domain* (Additional empirical evidence in Fig. 4A in App. G). We hypothesize that enforcing smoothness of classifier h_θ leads to a smooth landscape for discrepancy ($d_{S,T}^{\Phi}$) as it is also a function of h_θ. The smooth landscape ensures stable minimization (of Eq. 4), ensuring a decrease in ($d_{S,T}^{\Phi}$) with each SGD step even for a large step size (similar to [9]), this explains the enhanced stability and improved performance of domain adversarial training. For verifying the stabilization effect of smoothness, empirically we obtain $\lambda_{\max}$ for SGD and proposed SDAT at both best ($\lambda_{\max}^{best}$) and last epoch ($\lambda_{\max}^{last}$) for adaptation from Infographic to Clipart Domain (Fig. 2D). We find that as $\lambda_{\max}$ increases (decrease in smoothness of landscape), the training becomes unstable for SGD leading to a drop in validation accuracy. Whereas in the case of the proposed SDAT, the $\lambda_{\max}$ remains low across epochs, leading to stable and better validation accuracy curve. We provide more validation accuracy curves for adaptation tasks where we also observe a similar phenomenon in Fig. 5. To the best of our knowledge, our analysis of the effect of smoothness of task loss on the stability of DAT is novel.
We also find that SGD leads to low $\lambda_{\max}$ (high smoothness w.r.t. task loss) in comparison to Adam leading to better performance. This also explains the widespread usage of SGD for DAT algorithms [16, 34, 41], instead of Adam. More details about Hessian analysis is provided in App. D.

3.1. Smoothing Loss Landscape

In this section we first introduce the losses which are based on Sharpness Aware Minimization [15] (SAM). The basic idea of SAM is to find a smoother minima (i.e. low loss in ϵ neighborhood of θ) by using the following objective given formally below:

$$\min_{\theta} \max_{||\epsilon|| \leq \rho} L_{obj}(\theta + \epsilon) \tag{5}$$

here L_{obj} is the objective function to be minimized and $\rho \geq 0$ is a hyperparameter which defines the maximum norm for ϵ. Since finding the exact solution of inner maximization is hard, SAM maximizes the first order approximation:

$$\begin{aligned} \hat{\epsilon}(\theta) &\approx \arg\max_{||\epsilon|| \leq \rho} L_{obj}(\theta) + \epsilon^{\mathbf{T}} \nabla_\theta L_{obj}(\theta) \\ &= \rho \nabla_\theta L_{obj}(\theta) / ||\nabla_\theta L_{obj}(\theta)||_2 \end{aligned} \tag{6}$$

The $\hat{\epsilon}(\theta)$ is added to the weights θ. The gradient update for θ is then computed as $\nabla_\theta L_{obj}(\theta)|_{\theta+\hat{\epsilon}(\theta)}$. The above procedure can be seen as a generic smoothness enhancing formulation for any L_{obj}. We now analogously introduce the sharpness aware source risk for finding a smooth minima:

$$\max_{||\epsilon|| \leq \rho} R_S^l(h_{\theta+\epsilon}) = \max_{||\epsilon|| \leq \rho} \mathbb{E}_{x \sim P_S}[\, l(h_{\theta+\epsilon}(x), f(x))] \tag{7}$$

We also now define the sharpness aware discrepancy estimation objective below:

$$\max_{\Phi} \min_{||\epsilon|| \leq \rho} d_{S,T}^{\Phi+\epsilon} \tag{8}$$

As $d_{S,T}^{\Phi}$ is to be maximized the sharpness aware objective will have $\min_{||\epsilon|| \leq \rho}$ instead of $\max_{||\epsilon|| \leq \rho}$, as it needs to find smoother maxima.

We find that above smooth discrepancy (sharpness aware) based smooth discriminator, i.e. SDAT w/ adv results in sub-optimal generalization on target domain i.e. high target error $R_T^l(h_\theta)$ (Fig. 4B). We also observe that further increasing the smoothness of the discriminator (i.e. discrepancy) w.r.t. adversarial loss (increasing ρ) leads to lowering of performance on the target domain (Fig. 4C). We also show that for gradient Lipschitz functions the discriminator is suboptimal between source and target domain, with smooth version of adversarial loss (App. C). A similar trend is observed in GANs (App. E) which also has a similar min-max objective. The theoretical justification of the suboptimality of smooth discrepancy is provided in App. C.

3.2. Smooth Domain Adversarial Training (SDAT)

We propose Smooth Domain Adversarial Training which only focuses on converging to smooth minima w.r.t. task loss (i.e. empirical source risk), whereas preserves the original discrepancy term. We define the optimization objective of proposed Smooth Domain Adversarial Training below:

$$\min_{\theta} \max_{\Phi} \max_{||\epsilon|| \leq \rho} \mathbb{E}_{x \sim P_S}[l(h_{\theta+\epsilon}(x), y(x))] + d^{\Phi}_{S,T} \quad (9)$$

The first term is the sharpness aware risk, and the second term is the discrepancy term which is not smooth in our procedure. The term $d^{\Phi}_{S,T}$ estimates $D^{\phi}_{h_\theta,H}(P_S||P_T)$ discrepancy. Any DAT baseline can be modified to use SDAT objective just by using few lines of code (App. L). *We observe that the proposed SDAT objective (Eq. 9) leads to significantly lower generalization error compared to the original DA objective (Eq. 4), which we empirically demonstrate in the following sections.*

4. Adaptation for classification

We evaluate our proposed method on three datasets: Office-Home [49], VisDA-2017 [39], and DomainNet [39], as well as by combining SDAT with two DAT based DA techniques: CDAN and CDAN+MCC. We also show results with ViT backbone on Office-Home and VisDA-2017 dataset. We show the effectiveness of our method primarily on the popular domain adaptation method CDAN [34] and CDAN+MCC [27], where a minimum class confusion (MCC) loss term is added as a regularizer to CDAN. Additional information regarding the methods is given in App. F. Moreover, the application of SDAT in DA for object detection is given in App. M. Additional ablations on SDAT is discussed in App. N.

Table 1. Accuracy (%) on Office-Home and VisDA-2017 for unsupervised DA (with ResNet-50 and ViT backbone).

Method		Avg (Office-Home)	Avg (VisDA-2017)
CDAN	ResNet-50	68.4	80.6
CDAN w/ SDAT		69.5	82.1
CDAN + MCC		71.3	83.6
CDAN + MCC w/ SDAT		**72.2**	**84.3**
TVT [53]	ViT	83.6	83.9
CDAN		79.3	79.6
CDAN w/ SDAT		82.4	84.5
CDAN + MCC		82.2	87.7
CDAN + MCC w/ SDAT		**84.3**	**89.8**

4.1. Results

Office-Home & VisDA-2017 : Table 1 compiles the result of our method on benchmark datasets. CDAN+MCC w/ SDAT achieves SOTA adversarial adaptation performance on the Office-Home dataset with ResNet-50 backbone. With ViT backbone, the increase in accuracy due to SDAT is more significant. This may be attributed to the observation that ViTs reach a sharp minima compared to ResNets [7]. CDAN w/ SDAT improves over CDAN by 1.1% with ResNet-50 and 3.1% with ViT backbone on Office-Home dataset. A similar trend is seen for VisDA-2017. CDAN + MCC w/ SDAT outperforms TVT [53], a recent ViT based DA method and achieves SOTA results on both Office-Home and VisDA datasets. App. J covers additional comparison of the proposed method with TVT. Table 7 and Table 6 in App. G compiles the performance of the proposed method across all the source-target pairs of Office-Home and VisDA-2017 dataset, respectively.

DomainNet: Table 2 shows the results on the large and challenging DomainNet dataset across five domains. The proposed method improves the performance of CDAN significantly across all source-target pairs. On specific source-target pairs like inf→real, the performance increase is 4.5%. The overall performance of CDAN is improved by nearly 1.8% which is significant considering the large number of classes and images present in DomainNet. The improved results are attributed to stabilized domain adversarial training through proposed SDAT which can be seen in Fig. 2D.

Table 2. Results on DomainNet with CDAN w/ SDAT. The number in the parenthesis refers to the increase in Acc. w.r.t. CDAN.

Target (→) Source (↓)	clp	inf	pnt	real	skt	Avg
clp	-	22.0 (+1.4)	41.5 (+2.6)	57.5 (+1.5)	47.2 (+2.3)	42.1 (+2.0)
inf	33.9 (+2.3)	-	30.3 (+1.0)	48.1 (+4.5)	27.9 (1.5)	35.0 (+2.3)
pnt	47.5 (+3.4)	20.7 (+0.9)	-	58.0 (+0.8)	41.8 (+1.8)	42.0 (+1.7)
real	56.7 (+0.9)	25.1 (+0.7)	53.6 (+0.4)	-	43.9 (+1.6)	44.8 (+1.0)
skt	58.7 (+2.7)	21.8 (+1.1)	48.1 (+2.8)	57.1 (+2.2)	-	46.4 (+2.2)
Avg	49.2 (+2.3)	22.4 (+1.0)	43.4 (+1.7)	55.2 (+2.2)	40.2 (+1.8)	42.1 (+1.8)

5. Conclusion

In this work, we analyze the curvature of loss surface of DAT used extensively for Unsupervised DA. We find that converging to a smooth minima w.r.t. task loss (i.e., empirical source risk) leads to stable DAT which results in better generalization on the target domain. We also theoretically and empirically show that smoothing adversarial components of loss lead to sub-optimal results, hence should be avoided in practice. We then introduce our practical and effective method, SDAT, which only increases the smoothness w.r.t. task loss, leading to better generalization on the target domain. SDAT leads to an effective increase for the latest methods for adversarial DA, achieving SOTA performance on benchmark datasets.

6. Acknowledgement

This work was supported in part by SERB-STAR Project (Project:STR/2020/000128), Govt. of India. Harsh Rangwani is supported by Prime Minister's Research Fellowship (PMRF). We are thankful for their support.

References

[1] David Acuna, Guojun Zhang, Marc T Law, and Sanja Fidler. f-domain-adversarial learning: Theory and algorithms. *arXiv preprint arXiv:2106.11344*, 2021. 1, 2, 9, 11, 12, 15

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 2, 11

[3] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alex Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. *arXiv preprint arXiv:2106.04732*, 2021. 18

[4] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. 14

[5] Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184(1):71–120, 2020. 10

[6] Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *arXiv preprint arXiv:2102.08604*, 2021. 16, 21

[7] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pretraining or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021. 4, 12

[8] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 14, 20

[9] Casey Chu, Kentaro Minami, and Kenji Fukumizu. Smoothness and stability in gans. *arXiv preprint arXiv:2002.04185*, 2020. 3

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 20

[11] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Tian Qi. Gradually vanishing bridge for adversarial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 21

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 13

[13] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017. 1

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 20

[15] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. 1, 3, 11, 16, 21

[16] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 1, 3, 15

[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*, 17(1):2096–2030, 2016. 12, 13, 15, 21

[18] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019. 12

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[20] Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: beyond sharp and flat local minima. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 2553–2564, 2019. 1

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 15, 20

[22] Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. *Advances in neural information processing systems*, 7, 1994. 1

[23] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997. 1

[24] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5001–5009, 2018. 20

[25] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. 16

[26] Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho*, and Krzysztof Geras*. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020. 2

[27] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In

European Conference on Computer Vision, pages 464–480. Springer, 2020. 4, 12, 13, 15

[28] Bo Fu Junguang Jiang, Baixu Chen and Mingsheng Long. Transfer-learning-library. https://github.com/thuml/Transfer-Learning-Library, 2020. 13, 14, 18

[29] Minguk Kang and Jaesik Park. ContraGAN: Contrastive Learning for Conditional Image Generation. 2020. 12

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[31] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 12

[32] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R. Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7046–7056, October 2021. 1

[33] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, Rahul M V, and R. Venkatesh Babu. Towards inheritable models for open-set domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[34] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1645–1655, 2018. 1, 2, 3, 4, 12, 13, 15

[35] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 12

[36] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019. 16, 21

[37] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010. 2

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 13

[39] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 4

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 14, 20

[41] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. In *International Conference on Learning Representations*, 2018. 3

[42] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 20

[43] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 12

[44] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. 20

[45] David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. *arXiv preprint arXiv:2104.04448*, 2021. 16, 17, 21

[46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 16, 21

[47] Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8725–8735, 2020. 15

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

[49] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[50] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: towards improving transferability of deep neural networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 1953–1963, 2019. 1

[51] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 13

[52] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 14

[53] Jinyu Yang, Jingjing Liu, Ning Xu, and Junzhou Huang. Tvt: Transferable vision transformer for unsupervised domain adaptation. *arXiv preprint arXiv:2108.05988*, 2021. 4, 14, 15, 17

[54] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 581–590. IEEE, 2020. 12

[55] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging theory and algorithm for domain adaptation. In *International Conference on Machine Learning*, pages 7404–7413. PMLR, 2019. 1, 15

Contents

A. Notation Table

Table 3 contains all the notations used in the paper and the proofs of theorems.

Table 3. The notations used in the paper and the corresponding meaning.

Notation	Meaning
S	Labeled Source Data
T	Unlabelled Target Data
P_S (or P_T)	Source (or Target) Distribution
$\mathcal{X}$	Input space
$\mathcal{Y}$	Label space
$y(\cdot)$	Maps image to labels
h_θ	Hypothesis function
$R^l_S(h_\theta)$ (or $R^l_T(h_\theta)$)	Source (or Target) risk
$\hat{R}^l_S(h_\theta)$ (or $\hat{R}^l_T(h_\theta)$)	Empirical Source (or Target) risk
$\mathcal{H}$	Hypothesis space
$D^\phi_{h_\theta,\mathcal{H}}(P_S\|\|P_T)$	Discrepancy between two domains P_S and P_T
g_ψ	Feature extractor
f_Θ	Classifier
$\mathcal{D}_\Phi$	Domain Discriminator
$d^\Phi_{S,T}$	Tractable Discrepancy Estimate
$\nabla^2_\theta \hat{R}^l_S(h_\theta)$ (or H)	Hessian of classification loss
$Tr(H)$	Trace of Hessian
λ_{max}	Maximum eigenvalue of Hessian
ϵ	Perturbation
ρ	Maximum norm of ϵ

B. Connection of Discrepancy to $d^\Phi_{S,T}$ (Eq. 4) in Main Paper

We refer reader to Appendix C.2 of [1] for relation of $d^\Phi_{S,T}$. The $d^\Phi_{S,T}$ term defined in Eq. 4 given as:

$$d^\Phi_{S,T} = \mathbb{E}_{x\sim P_S}[\log(\mathcal{D}_\Phi(g_\psi(x)))] + \mathbb{E}_{x\sim P_T}[\log(1-\mathcal{D}_\Phi(g_\psi(x)))] \tag{10}$$

The above term is exactly the Eq. C.1 in [1] where they show that optimal $d^\Phi_{S,T}$ i.e.:

$$\max_\Phi d^\Phi_{S,T} = D_{JS}(P_S||P_T) - 2\log(2) \tag{11}$$

Hence we can say from result in Eq. 4 is a consequence of Lemma 1 and Proposition 1 in [1], assuming that D_Φ satisfies the constraints in Proposition 1.

C. Theoretical Analysis

In this section, we present theoretical results and their corresponding proofs for the proposed SDAT.

Theorem 1 (Generalization bound). *Suppose $l : \mathcal{Y}\times\mathcal{Y} \to [0,1] \subset dom\ \phi^*$. Let h^* be the ideal joint classifier with error $\lambda^* = R^l_S(h^*) + R^l_T(h^*)$. We have the following relation between source and target risk:*

$$R^l_T(h_\theta) \le R^l_S(h_\theta) + D^\phi_{h_\theta,\mathcal{H}}(P_S||P_T) + \lambda^* \tag{12}$$

The above generalization bound shows that the target risk $R^l_T(h_\theta)$ is upper bounded by the source risk $R^l_S(h_\theta)$ and the discrepancy term $D^\phi_{h_\theta,\mathcal{H}}$ along with an irreducible constant error λ^*. Hence, this infers that reducing source risk and discrepancy lead a to reduction in target risk.

Proof. We refer the reader to Theorem 2 in Appendix B of [1] for the detailed proof the theorem. □

We now introduce a Lemma for smooth functions which we will use in the proofs subsequently:

Lemma 1. *For an L-smooth function $f(w)$ the following holds where w^* is the optimal minima:*

$$f(w) - f(w^*) \geq \frac{1}{2L}||\nabla f(w)||^2$$

Proof. The L-smooth function by definition satisfies the following:

$$f(w^*) \leq f(v) \leq f(w) + \nabla f(w)(v - w) + \frac{L}{2}||v - w||^2$$

Now we minimize the upper bound wrt v to get a tight bound on $f(w^*)$.

$$D(v) = f(w) + \nabla f(w)(v - w) + \frac{L}{2}||v - w||^2$$

after doing $\nabla_v D(v) = 0$ we get:

$$v = w - \frac{1}{L}\nabla f(w)$$

By substituting the value of v in the upper bound we get:

$$f(w^*) \leq f(w) - \frac{1}{2L}||\nabla f(w)||^2$$

Hence rearranging the above term gives the desired result:

$$f(w) - f(w^*) \geq \frac{1}{2L}||\nabla f(w)||^2$$

□

We now theoretically analyze the difference in discrepancy estimation for smooth version $d^{\Phi''}_{S,T}$ (Eq. 8) in comparison to non-smooth version $d^{\Phi'}_{S,T}$ (Eq. 3). Assuming $\mathcal{D}_\Phi$ is a L-smooth function (common assumption for non-convex optimization [5]), η is a small constant and $d^*_{S,T}$ the optimal discrepancy, the theorem states:

Theorem 2. *For a given classifier h_θ and one step of (steepest) gradient ascent i.e. $\Phi' = \Phi + \eta(\nabla d^\Phi_{S,T}/||\nabla d^\Phi_{S,T}||)$ and $\Phi'' = \Phi + \eta(\nabla d^\Phi_{S,T}|_{\Phi+\hat{\epsilon}(\Phi)}/||\nabla d^\Phi_{S,T}|_{\Phi+\hat{\epsilon}(\Phi)}||)$ for maximizing*

$$d^{\Phi'}_{S,T} - d^{\Phi''}_{S,T} \leq \eta(1 - \cos\alpha)\sqrt{2L(d^*_{S,T} - d^\Phi_{S,T})} \tag{13}$$

where α is the angle between $\nabla d^\Phi_{S,T}$ and $\nabla d^\Phi_{S,T}|_{\Phi+\hat{\epsilon}(\Phi)}$.

The $d^{\Phi'}_{S,T}$ (non-smooth version) can exceed $d^{\Phi''}_{S,T}$ (smooth discrepancy) significantly, as the term $d^*_{S,T} - d^\Phi_{S,T} \not\to 0$, as the h_θ objective is to oppose the convergence of $d^\Phi_{S,T}$ to optima $d^*_{S,T}$ (min-max training in Eq. 4). Thus $d^{\Phi'}_{S,T}$ can be a better estimate of discrepancy in comparison to $d^{\Phi''}_{S,T}$. A better estimate of $d^\Phi_{s,t}$ helps in effectively reducing the discrepancy between P_S and P_T, hence leads to reduced $R^l_T(h_\theta)$.

Proof of Theorem 2. We assume that the function is L-smooth (the assumption of L-smoothness is the basis of many results in non-convex optimization [5]) in terms of input x. As for a fixed h_θ as we use a reverse gradient procedure for measuring the discrepancy, only one step analysis is shown. This is because only a single step of gradient is used for estimating discrepancy $d^\Phi_{S,T}$ i.e. one step of each min and max optimization is performed alternatively for optimization. After this the h_θ is updated to decrease the discrepancy. Any differential function can be approximated by the linear approximation in case of small η:

$$d^{\Phi+\eta v}_{S,T} \approx d^\Phi_{S,T} + \eta \nabla d^{\Phi^\mathbf{T}}_{S,T} v \tag{14}$$

The dot product between two vectors can be written as the following function of norms and angle θ between those:

$$\nabla d^{\Phi^\mathbf{T}}_{S,T} v = ||\nabla d^\Phi_{S,T}||\ ||v||\ cos\theta \tag{15}$$

The steepest value will be achieved when $\cos\theta = 1$ which is actually $v = \frac{\nabla d^{\Phi}_{S,T}(x)}{||\nabla d^{\Phi}_{S,T}(x)||}$. Now we compare the descent in another direction $v_2 = \frac{\nabla d^{\Phi}_{S,T}|_{w+\epsilon(w)}}{||\nabla d^{\Phi}_{S,T}|_{w+\epsilon(w)}||}$ from the gradient descent. The difference in value can be characterized by:

$$d^{\Phi+\eta v}_{S,T} - d^{\Phi+\eta v_2}_{S,T} = \eta||\nabla d^{\Phi}_{S,T}||(1-\cos\alpha) \tag{16}$$

As α is an angle between $\nabla d^{\Phi}_{S,T}|_{w+\epsilon(w)}$ (v_2) and $\nabla d^{\Phi}_{S,T}(X)$ (v). The suboptimality is dependent on the gradient magnitude. We use the following result to show that when optimality gap $d^{*}_{S,T} - d^{\Phi}_{S,T}(x)$ is large the difference between two directions is also large.

For an L-smooth function the following holds according to Lemma 1:

$$f(w) - f(w^*) \geq \frac{1}{2L}||\nabla f(w)||^2$$

As we are performing gradient ascent $f(w) = -d^{\Phi}_{s,t}$, we get the following result:

$$(d^{*}_{S,T} - d^{\Phi}_{S,T}) \geq \frac{1}{2L}||\nabla d^{\Phi}_{S,T}(x)||^2$$

$$2L(d^{*}_{S,T} - d^{\Phi}_{S,T}) \geq \frac{(d^{\Phi+\eta v_2}_{S,T} - d^{\Phi+\eta v}_{S,T})^2}{(\eta(1-\cos\alpha))^2}$$

$$\eta(1-\cos\alpha)\sqrt{2L(d^{*}_{S,T} - d^{\Phi}_{S,T})} \geq (d^{\Phi'}_{S,T} - d^{\Phi''}_{S,T})$$

This shows that difference in value of by taking a step in direction of gradient v vs taking the step in a different direction v_2 is upper bounded by the $d^{*}_{S,T} - d^{\Phi}_{S,T}(x)$, hence if we are far from minima the difference can be potentially large. As we are only doing one step of gradient ascent $d^{*}_{S,T} - d^{\Phi}_{S,T}$ will be potentially large, hence can lead to suboptimal measure of discrepancy. □

We now show that optimizing Eq. 9 reduces $R^l_T(h_\theta)$ through a generalization bound. This bound establishes that our proposed SDAT procedure is also consistent (i.e. in case of infinite data the upper bound is tight) similar to the original DAT objective (Eq. 4).

Theorem 3. *Suppose l is the loss function, we denote $\lambda^* := R^l_S(h^*) + R^l_T(h^*)$ and let h^* be the ideal joint hypothesis:*

$$R^l_T(h_\theta) \leq \max_{||\epsilon||\leq\rho} \hat{R}^l_S(h_{\theta+\epsilon}) + D^{\phi}_{h_\theta,H}(P_S||P_T) + \gamma(||\theta||^2_2/\rho^2) + \lambda^*. \tag{17}$$

where $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ is a strictly increasing function.

The bound is similar to generalization bounds for domain adaptation [1, 2]. The main difference is the sharpness aware risk term $\max_{||\epsilon||\leq\rho} \hat{R}^l_S(h_\theta)$ in place of source risk $R^l_S(h_\theta)$, and an additional term that depends on the norm of the weights $\gamma(||\theta||^2_2/\rho^2)$. The first is minimized by decreasing the empirical sharpness aware source risk by using SAM loss shown in Sec. 3. The second term is reduced by decreasing the discrepancy between source and target domains. The third term, as it is a function of norm of weights $||\theta||^2_2$, can be reduced by using either L2 regularization or weight decay. Since we assume that the $\mathcal{H}$ hypothesis class we have is rich, the λ^* term is small.

Proof of Theorem 3: In this case we make use of Theorem 2 in the paper sharpness aware minimization [15] which states the following: The source risk $R_S(h)$ is bounded using the following PAC-Bayes generalization bound for any ρ with probability $1-\delta$:

$$R_S(h_\theta) \leq \max_{||\epsilon||\leq\rho} \hat{R}_S(h_\theta) + \sqrt{\frac{k\log\left(1 + \frac{\|\boldsymbol{\theta}\|^2_2}{\rho^2}\left(1+\sqrt{\frac{\log(n)}{k}}\right)^2\right) + 4\log\frac{n}{\delta} + \tilde{O}(1)}{n-1}} \tag{18}$$

Table 4. Architecture used for feature classifier and Domain classifier. C is the number of classes. Both classifiers will take input from feature generator (g_θ).

Layer	Output Shape
Feature Classifier (f_Θ)	
-	Bottleneck Dimension
Linear	C
Domain Classifier ($\mathcal{D}_\Phi$)	
-	Bottleneck Dimension
Linear	1024
BatchNorm	1024
ReLU	1024
Linear	1024
BatchNorm	1024
ReLU	1024
Linear	1

Table 5. Accuracy (%) on VisDA-2017 (ResNet-101 and ViT backbone).

Method		**Synthetic → Real**
DANN [17]	ResNet-101	57.4
MCD [43]		71.4
CDAN* [34]		73.7
CDAN		76.6
CDAN w/ SDAT		78.3
CDAN+MCC [27]		80.4
CDAN+MCC w/ SDAT		**81.2**
CDAN	ViT	76.7
CDAN w/ SDAT		81.1
CDAN+MCC [27]		85.1
CDAN+MCC w/ SDAT		**87.8**

here n is the training set size used for calculation of empirical risk $\hat{R}_S(h)$, k is the number of parameters and $||\theta||_2$ is the norm of the weight parameters. The second term in equation can be abbreviated as $\gamma(||\theta||_2)$. Hence,

$$R_S(h_\theta) \leq \max_{||\epsilon||\leq\rho} \hat{R}_S(h_\theta) + \gamma(||\theta||_2^2/\rho^2) \tag{19}$$

From the generalization bound for domain adaptation for any f-divergence [1] (Theorem 2) we have the following result.

$$R_T^l(h_\theta) \leq R_S^l(h_\theta) + \mathcal{D}^\phi_{h_\theta,H}(P_S||P_T) + \lambda^* \tag{20}$$

Combining the above two inequalities gives us the required result we wanted to prove i.e.

$$R_T^l(h_\theta) \leq \tilde{R}_S^l(h_\theta) + D^\phi_{h_\theta,H}(P_S||P_T) + \gamma(||\theta||_2^2/\rho^2) + \lambda^*. \tag{21}$$

□

D. Hessian Analysis

We use the PyHessian library [54] to calculate the Hessian eigenvalues and the Hessian Eigen Spectral Density. For Office-Home experiments, all the calculations are performed using 50% of the source data at the last checkpoint. For DomainNet experiments (Fig. 2D), we use 10% of the source data for Hessian calculation. The Maximum Eigenvalue is calculated at the checkpoint with the best validation accuracy ($\lambda_{\max}^{best}$) and the last checkpoint ($\lambda_{\max}^{last}$). Only the source class loss is used for calculating to clearly illustrate our point. The partition was selected randomly, and the same partition was used across all the runs. We also made sure to use the same environment to run all the Hessian experiments. A subset of the data was used for Hessian calculation mainly because the hessian calculation is computationally expensive [54]. This is commonly done in hessian experiments. For example, [7] (refer Appendix D) uses 10% of training data for Hessian Eigenvalue calculation. The PyHessian library uses Lanczos algorithm [18] for calculating the Eigen Spectral density of the Hessian and uses the Hutchinson method to calculate the trace of the Hessian efficiently.

E. Smoothness of Discriminator in SNGAN

For further establishing the generality of sub-optimality of smooth adversarial loss, we also perform experiments on Spectral Normalised Generative Adversarial Networks (SNGAN) [35]. In case of SNGAN we also find that smoothing discriminator through SAM leads to suboptimal performance (higher FID) as in Fig. 3. The above evidences indicates that *smoothing the adversarial loss leads to sub-optimality*, hence it should not be done in practice. We use the same configuration for SNGAN as described in PyTorchStudioGAN [29] for both CIFAR10 [31] and TinyImageNet [2] with batch size of 256 in both cases. We then smooth the discriminator while discriminator is trained by using the same formulation as in Eq. 8. We find that smoothing discriminator leads to higher (suboptimal) Fréchet Inception Distance in case of GANs as well, shown in Fig. 3.

[2]https://www.kaggle.com/c/tiny-imagenet

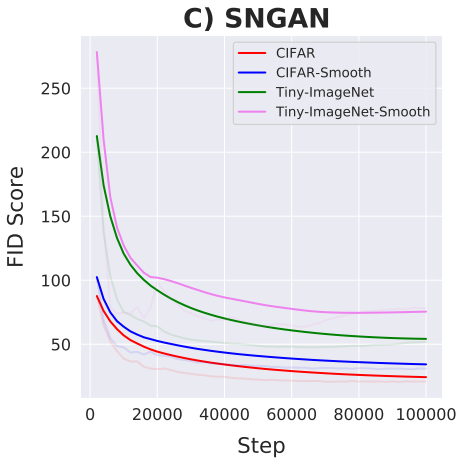


Figure 3. SNGAN performance on different datasets, smoothing discriminator in GAN also leads to inferior GAN performance (higher FID) across both datasets.

F. Experimental Details

F.1. Image Classification

We primarily showcase the efficacy of our proposed method by implementing it on the following domain adaptation methods:

- **CDAN** [34]: Conditional Domain Adversarial network is a popular DA algorithm that improves the performance of the DANN algorithm. CDAN introduces the idea of multi-linear conditioning to align the source and target distributions better. CDAN in Table 7 and 6 refers to our implementation of CDAN* [34] method.

- **CDAN + MCC** [27]: The minimum class confusion (MCC) loss term is added as a regularizer to CDAN. MCC is a non-adversarial term that minimizes the pairwise class confusion on the target domain, hence we consider this as an additional minimization term which is added to empirical source risk. This method achieves close to SOTA accuracy among adversarial adaptation methods.

We implement our proposed method in the Transfer-Learning-Library [28] toolkit developed in PyTorch [38]. The difference between the performance reported in CDAN* and our implementation CDAN is due to the batch normalization layer in domain classifier, which enhances performance. We tune ρ value in SDAT for a particular dataset split and use the same value across domains. The ρ value is set to 0.02 for the Office-Home experiments, 0.005 for the VisDA-2017 experiments and 0.05 for the DomainNet experiments.

Office-Home: For CDAN methods with ResNet-50 backbone, we train the models using mini-batch stochastic gradient descent (SGD) with a batch size of 32 and a learning rate of 0.01. The learning rate schedule is the same as [17]. We train it for a total of 30 epochs with 1000 iterations per epoch. The momentum parameter in SGD is set to 0.9 and a weight decay of 0.001 is used. For CDAN+MCC experiments with ResNet-50 backbone, we use a temperature parameter [27] of 2.5. The bottleneck dimension for the features is set to 2048. The difference between the performance reported in CDAN* and our implementation CDAN is due to the batch normalization layer in domain classifier, which enhances performance.

VisDA-2017: We use a ResNet-101 backbone initialized with ImageNet weights for VisDA-2017 experiments. Center Crop is also used as an augmentation during training. We use a bottleneck dimension of 256 for both algorithms. For CDAN runs, we train the model for 30 epochs with same optimizer setting as that of Office-Home. For CDAN+MCC runs, we use a temperature parameter of 3.0 and a learning rate of 0.002.

DomainNet: We use a ResNet-101 backbone initialized with ImageNet weights for DomainNet experiments. We run all the experiments for 30 epochs with 2500 iterations per epoch. The other parameters are the same as that of Office-Home.

Additional experiments with a ViT backbone are performed on Office-Home and VisDA-2017 datasets. We use the ViT-B/16 [12] architecture pretrained on ImageNet-1k, the implementation of which is borrowed from [51]. For all CDAN runs on Office-Home and VisDA, we use an initial learning rate of 0.01, whereas for CDAN+MCC runs, the initial learning rate

of 0.002 is used. ρ value of 0.02 is shared across all the splits on both the datasets for the ViT backbone. A batch-size of 24 is used for Office-Home and 32 for VisDA-2017.

To show the effectiveness of SDAT fairly and promote reproducibility, we run with and without SDAT on the same GPU and environment and with the same seed. All the above experiments were run on Nvidia V100, RTX 2080 and RTX A5000 GPUs. We used Wandb [4] to track our experiments. We will be releasing the code to promote reproducible research.

F.1.1 Architecture of Domain Discriminator

One of the major reasons for increased accuracy in Office-Home baseline CDAN compared to reported numbers in the paper is the architecture of domain classifier. The main difference is the use of batch normalization layer in domain classifier, which was done in the library [28]. Table 4 shows the architecture of the feature classifier and domain classifier.

F.2. Additional Implementations Details for DA for Object detection

In SDAT, we modified the loss function present in [8] by adding classification loss smoothing, i.e. smoothing classification loss of RPN and ROI, used in Faster R-CNN [40], by training with source data. Similarly, we applied smoothing to regression loss and found it to be less effective. We implemented SDAT for object detection using Detectron2 [52]. The training is done via SGD with momentum 0.9 for 70k iterations with the learning rate of 0.001, and then dropped to 0.0001 after 50k iterations. We split the target data into train and validation sets and report the best mAP on validation data. We fixed ρ to 0.15 for object detection experiments.

G. Additional Results

Table 6. Accuracy (%) on VisDA-2017 for unsupervised DA (with ResNet-101 and ViT backbone). The **mean** column contains mean across all classes. SDAT particularly improves the accuracy in classes that have comparatively low CDAN performance.

Method		**plane**	**bcycl**	**bus**	**car**	**horse**	**knife**	**mcyle**	**persn**	**plant**	**sktb**	**train**	**truck**	**mean**
CDAN	ResNet-101	94.9	72.0	83.0	57.3	91.6	95.2	91.6	79.5	85.8	88.8	87.0	40.5	80.6
CDAN w/ SDAT	ResNet-101	94.8	77.1	82.8	60.9	92.3	95.2	**91.7**	**79.9**	89.9	91.2	**88.5**	41.2	82.1
CDAN+MCC	ResNet-101	95.0	84.2	75.0	66.9	**94.4**	97.1	90.5	79.8	89.4	89.5	86.9	54.4	83.6
CDAN+MCC w/ SDAT	ResNet-101	**95.8**	**85.5**	76.9	69.0	93.5	**97.4**	88.5	78.2	**93.1**	**91.6**	86.3	**55.3**	**84.3**
TVT [53]	ViT	92.9	85.6	77.5	60.5	93.6	98.2	89.3	76.4	93.6	92.0	91.7	55.7	83.9
CDAN	ViT	94.3	53.0	75.7	60.5	93.9	98.3	**96.4**	77.5	91.6	81.8	87.4	45.2	79.6
CDAN w/ SDAT	ViT	96.3	80.7	74.5	65.4	95.8	**99.5**	92.0	83.7	93.6	88.9	85.8	57.2	84.5
CDAN+MCC	ViT	96.9	89.8	82.2	74.0	96.5	98.5	95.0	81.5	95.4	92.5	91.4	**58.5**	87.7
CDAN+MCC w/ SDAT	ViT	**98.4**	**90.9**	**85.4**	**82.1**	**98.5**	97.6	96.3	**86.1**	**96.2**	**96.7**	**92.9**	56.8	**89.8**

VisDA-2017: CDAN w/ SDAT improves the overall performance of CDAN by more than 1.5% with ResNet backbone and by 4.9% with ViT backbone on VisDA-2017 dataset (Table 6). Also, on CDAN + MCC baseline SDAT leads to 2.1% improvement over baseline, leading to SOTA accuracy of 89.8% across classes. Particularly, SDAT significantly improves the performance of underperforming minority classes like bicycle, car and truck. Additional discussion on statistical significance (App. K).

Table 5 shows the overall accuracy on the VisDA-2017 with ResNet-101 and ViT backbone. The accuracy reported in this table is the overall accuracy of the dataset, whereas the accuracy reported in the Table Table 6 refers to the mean of the accuracy across classes. CDAN w/ SDAT outperforms CDAN by 1.7% with ResNet-101 and by 4.4% with ViT backbone, showing the effectiveness of SDAT in large scale Synthetic $\rightarrow$ Real shifts. With CDAN+MCC as the DA method, adding SDAT improves the performance of the method to 81.2% with ResNet-101 backbone.

Office-Home: Table 7 compiles the accuracy on all the splits of Office-Home dataset. We also compare our method with other DA algorithms including DANN, SRDC, MDD and f-DAL. The addition of SDAT improves the performance on both CDAN and CDAN+MCC across majority of source and target domain pairs. CDAN+MCC w/ SDAT with ViT backbone outperforms other SOTA DA techniques . With ViT backbone, SDAT particularly improves the performance of source-target pairs which have low accuracy on the target domain (Pr→Cl, Rw→Cl, Pr→Ar, Ar→Pr.

Table 7. Accuracy (%) on Office-Home for unsupervised DA (with ResNet-50 and ViT backbone). CDAN+MCC w/ SDAT outperforms other SOTA DA techniques. CDAN w/ SDAT improves over CDAN by 1.1% with ResNet-50 and 3.1% with ViT backbone.

Method		**Ar→Cl**	**Ar→Pr**	**Ar→Rw**	**Cl→Ar**	**Cl→Pr**	**Cl→Rw**	**Pr→Ar**	**Pr→Cl**	**Pr→Rw**	**Rw→Ar**	**Rw→Cl**	**Rw→Pr**	**Avg**
ResNet-50 [21]	ResNet-50	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [17]		45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN* [34]		49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
MDD [55]		54.9	73.7	77.8	60.0	71.4	71.8	61.2	53.6	78.1	72.5	60.2	82.3	68.1
f-DAL [1]		56.7	77.0	81.1	63.1	72.2	75.9	64.5	54.4	81.0	72.3	58.4	83.7	70.0
SRDC [47]		52.3	76.3	81.0	**69.5**	76.2	**78.0**	**68.7**	53.8	81.7	**76.3**	57.1	85.0	71.3
CDAN		54.3	70.6	76.8	61.3	69.5	71.3	61.7	55.3	80.5	74.8	60.1	84.2	68.4
CDAN w/ SDAT		56.0	72.2	78.6	62.5	73.2	71.8	62.1	55.9	80.3	75.0	61.4	84.5	69.5
CDAN + MCC		57.0	76.0	81.6	64.9	75.9	75.4	63.7	56.1	81.2	74.2	63.9	85.4	71.3
CDAN + MCC w/ SDAT		**58.2**	**77.1**	**82.2**	66.3	**77.6**	76.8	63.3	**57.0**	**82.2**	74.9	**64.7**	**86.0**	**72.2**
TVT [53]	ViT	**74.9**	86.8	89.5	82.8	**87.9**	88.3	79.8	**71.9**	90.1	85.5	74.6	90.6	83.6
CDAN		62.6	82.9	87.2	79.2	84.9	87.1	77.9	63.3	88.7	83.1	63.5	90.8	79.3
CDAN w/ SDAT		69.1	86.6	88.9	81.9	86.2	88.0	81.0	66.7	89.7	86.2	72.1	91.9	82.4
CDAN + MCC		67.0	84.8	90.2	83.4	87.3	89.3	80.7	64.4	90.0	86.6	70.4	91.9	82.2
CDAN + MCC w/ SDAT		70.8	**87.0**	**90.5**	**85.2**	87.3	**89.7**	**84.1**	70.7	**90.6**	**88.3**	**75.5**	**92.1**	**84.3**

Table 8. Accuracy(%) on **DomainNet** dataset for unsupervised domain adaptation (ResNet-101) across five distinct domains. The row indicates the source domain and the columns indicate the target domain.

ADDA	clp	inf	pnt	rel	skt	Avg	MCD	clp	inf	pnt	rel	skt	Avg
clp	-	11.2	24.1	41.9	30.7	27.0	clp	-	14.2	26.1	45.0	33.8	29.8
inf	19.1	-	16.4	26.9	14.6	19.2	inf	23.6	-	21.2	36.7	18.0	24.9
pnt	31.2	9.5	-	39.1	25.4	26.3	pnt	34.4	14.8	-	50.5	28.4	32.0
rel	39.5	14.5	29.1	-	25.7	27.2	rel	42.6	19.6	42.6	-	29.3	33.5
skt	35.3	8.9	25.2	37.6	-	26.7	skt	41.2	13.7	27.6	34.8	-	29.3
Avg	31.3	11.0	23.7	36.4	24.1	25.3	Avg	35.4	15.6	29.4	41.7	27.4	29.9
CDAN	clp	inf	pnt	rel	skt	Avg	**CDAN w/ SDAT**	clp	inf	pnt	rel	skt	Avg
clp	-	20.6	38.9	56.0	44.9	40.1	clp	-	22.0	41.5	57.5	47.2	42.1
inf	31.5	-	29.3	43.6	26.3	32.7	inf	33.9	-	30.3	48.1	27.9	35.0
pnt	44.1	19.8	-	57.2	39.9	40.2	pnt	47.5	20.7	-	58.0	41.8	42.0
rel	55.8	24.4	53.2	-	42.3	43.9	rel	56.7	25.1	53.6	-	43.9	44.8
skt	56.0	20.7	45.3	54.9	-	44.2	skt	58.7	21.8	48.1	57.1	-	46.4
Avg	46.9	21.4	41.7	52.9	38.3	40.2	Avg	49.2	22.4	43.4	55.2	40.2	**42.1**

DomainNet: Table 8 shows the results of the proposed method on DomainNet across five domains. We compare our results with ADDA and MCD and show that CDAN achieves much higher performance on DomainNet compared to other techniques. It can be seen that CDAN w/ SDAT further improves the overall accuracy on DomainNet by 1.8%.
We have shown results with three different domain adaptation algorithms namely DANN [16], CDAN [34] and CDAN+MCC [27]. SDAT has shown to improve the performance of all the three DA methods. This shows that SDAT is a generic method that can applied on top of any domain adversarial training based method to get better performance.

Source-only: Source-only setting measures the performance of a model trained only on source domain directly on unseen target data with no further target adaptation. We compare the performance of models with and without smoothing the loss landscape for source-only experiments on VisDA-2017 (Table 9) and Office-Home (Table 10) datasets with a ViT backbone pretrained on ImageNet. Initial learning rate of 0.001 and 0.002 is used for Office-Home and VisDA-2017 dataset, respectively. ρ value of 0.002 is used for ERM w/SAM run for both the datasets. It can be seen that ERM w/ SAM does not directly lead to better performance on the target domain.

Table 9. Accuracy (%) of source-only model trained with SGD (ERM) and SAM (ERM w/SAM) on VisDA-2017 for unsupervised DA with ViT-B/16 backbone

Method	**plane**	**bcybl**	**bus**	**car**	**horse**	**knife**	**mcyle**	**persn**	**plant**	**sktb**	**train**	**truck**	**mean**
ERM	98.4	58.3	80.2	60.7	89.3	53.6	88.4	40.8	62.8	87.4	94.7	19.1	69.5
ERM w/ SAM	98.6	33.1	80.0	76.9	90.1	35.9	94.2	22.8	77.8	89.0	95.3	11.6	67.1

Table 10. Accuracy (%) of source-only model trained with SGD (ERM) and SAM (ERM w/SAM) on Office-Home for unsupervised DA with ViT-B/16 backbone

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ERM	51.5	80.8	86.0	74.8	80.2	82.6	71.8	51.0	85.5	79.5	55.0	87.9	73.9
ERM w/ SAM	50.8	79.5	85.2	72.6	78.4	81.4	71.8	49.6	85.2	79.0	52.8	87.2	72.8

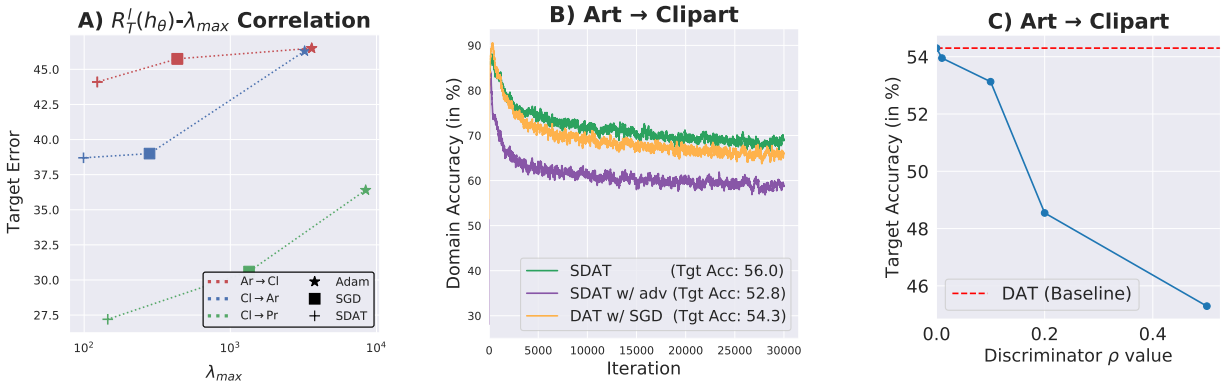


Figure 4. **A)** Error on Target Domain (y-axis) for Office-Home dataset against maximum eigenvalue λ_{max} of classification loss in DAT. When compared to SGD, Adam converges to a non-smooth minima (high λ_{max}), leading to a high error on target. Using Adam in comparison to SGD, converges to a non-smooth minima (high λ_{max}) leading to high error on target. **B)** Domain Accuracy (vs iterations), it is lower when adversarial loss is smooth (i.e. SDAT w/ adv), which indicates suboptimal discrepancy estimation $d^{\Phi}_{s,t}$ **C)** Target Accuracy on Art → Clipart vs smoothness of the adversarial component. As the smoothness increases (ρ), the target accuracy decreases indicating that smoothing adversarial loss leads to sub-optimal generalization.

H. Different Smoothing Techniques

Stochastic Weight Averaging (SWA) [25]: SWA is a widely popular technique to reach a flatter minima. The idea behind SWA is that averaging weights across epochs leads to better generalization because it reaches a wider optima. The recently proposed SWA-Densely (SWAD) [6] takes this a step further and proposes to average the weights across iterations instead of epochs. SWAD shows improved performance on domain generalization tasks. We average every 400 iterations in the SWA instead of averaging per epochs. We tried averaging across 800 iterations as well and the performance was comparable.
Difference between SWAD and SDAT: As SWAD performs Weight Averaging, it is not possible to selectively smooth only minimization (ERM) components with SWAD, as gradients for both the adversarial loss and ERM update weights of the backbone. Due to this, SWAD cannot reach optimal performance for DAT. For verifying this, we also compare our method by implementing SWAD for Domain Adaptation on four different source-target pairs of Office-Home dataset in Table 18. On average, SDAT (Ours) gets 61.6% (+2.4% over DAT) accuracy in comparison to 60.4% (+1.2% over DAT) for SWAD.
Virtual Adversarial Training (VAT) [36]: VAT is regularization technique which makes use of adversarial perturbations. Adversarial perturbations are created using Algo. 1 present in [36]. We added VAT by optimizing the following objective:

$$\min_{\theta} \mathbb{E}_{x\sim P_S}[\max_{||r||\leq\epsilon} D_{KL}(h_\theta(x)||h_\theta(x+r))] \tag{22}$$

This value acts as a negative measure of smoothness and minimizing this will make the model smooth. For training, we set hyperparameters ϵ to 15.0, ξ to 1e-6, and α as 0.1.
Label Smoothing (LS) [46]: The idea behind label smoothing is to have a distribution over outputs instead of one hot vectors. Assuming that there are k classes, the correct class gets a probability of 1 - α and the other classes gets a probability of $\alpha/(k-1)$. [45] mention that label smoothing tends to avoid sharper minima during training. We use a smoothing parameter (α) of 0.1 in all the experiments in Table 11. We also show results with smoothing parameter of 0.2 and observe comparable performance. We observe that label smoothing slightly improves the performance over DAT.
SAM [15]: In this method, we apply SAM directly to both the task loss and adversarial loss with $\rho = 0.05$ as suggested in the paper. It can be seen that the performance improvement of SAM over DAT is minimal, thus indicating the need for SDAT.

Table 11. Different Smoothing techniques. We refer to [45] to compare the proposed SDAT with other techniques to show the efficacy of SDAT. It can be seen that SDAT outperforms the other smoothing techniques significantly. Other smoothing techniques improve upon the performance of DAT showing that smoothing is indeed necessary for better adaptation.

Method	Ar→Cl	Cl→Pr	Rw→Cl	Pr→Cl
DAT	54.3	69.5	60.1	55.3
VAT	54.6	70.7	60.8	54.4
SWAD-400	54.6	71.0	60.9	55.2
LS ($\alpha = 0.1$)	53.6	71.6	59.9	53.4
LS ($\alpha = 0.2$)	53.5	71.2	60.5	53.2
SDAT	**55.9**	**73.2**	**61.4**	**55.9**

I. Optimum ρ value

Table 12 and 13 show that $\rho = 0.02$ works robustly across experiments providing an increase in performance (although it does not achieve the best result each time) and can be used as a rule of thumb.

Table 12. ρ value for DomainNet

Split	DAT	SDAT($\rho = 0.02$)	SDAT - Reported ($\rho = 0.05$)
clp→skt	44.9	46.7	47.2
skt→clp	56.0	59.0	58.7
skt→pnt	45.3	47.8	48.1
inf→rel	43.6	47.3	48.1

Table 13. ρ value for VisDA-2017 Synthetic → Real

Backbone	DAT	SDAT ($\rho = 0.02$)	SDAT Reported($\rho = 0.005$)
CDAN	76.6	78.2	78.3
CDAN+MCC	80.4	80.9	81.2

J. Comparison with TVT

TVT [53] is a recent work that reports performance higher than the other contemporary unsupervised DA methods on the publicly available datasets. This method uses a ViT backbone and focuses on exploiting the intrinsic properties of ViT to achieve better results on domain adaptation. Like us, TVT uses an adversarial method for adaptation to perform well on the unseen target data. On the contrary, they introduce additional modules within their architecture. The Transferability Adaption Module (TAM) is introduced to assist the ViT backbone in capturing both discriminative and transferable features. Additionally, the Discriminative Clustering Module (DCM) is used to perform discriminative clustering to achieve diverse and clustered features.

Even without using external modules to promote the transferability and discriminability in the features learned using ViT, we are able to report higher numbers than TVT. This advocates our efforts to show the efficacy of converging to a smooth minima w.r.t. task loss to achieve better domain alignment. Moreover, TVT uses a batch size of 64 to train the network, causing a memory requirement of more than 35GB for efficient training, which is significantly higher than the 11.5GB memory used by our method on a batch-size of 24 for Office-Home to obtain better results. This allows our method to be trained using a standard 12GB GPU, removing the need of an expensive hardware. The ViT backbone used by TVT is pretrained on a much larger ImageNet-21k dataset, whereas we use the backbone pretrained on ImageNet-1k dataset.

K. Significance and Stability of Empirical Results

To establish the empirical results' soundness and reliability, we run a subset of experiments (representative of each different source domain) on DomainNet. The experiments are repeated with three different random seeds leading to overall 36

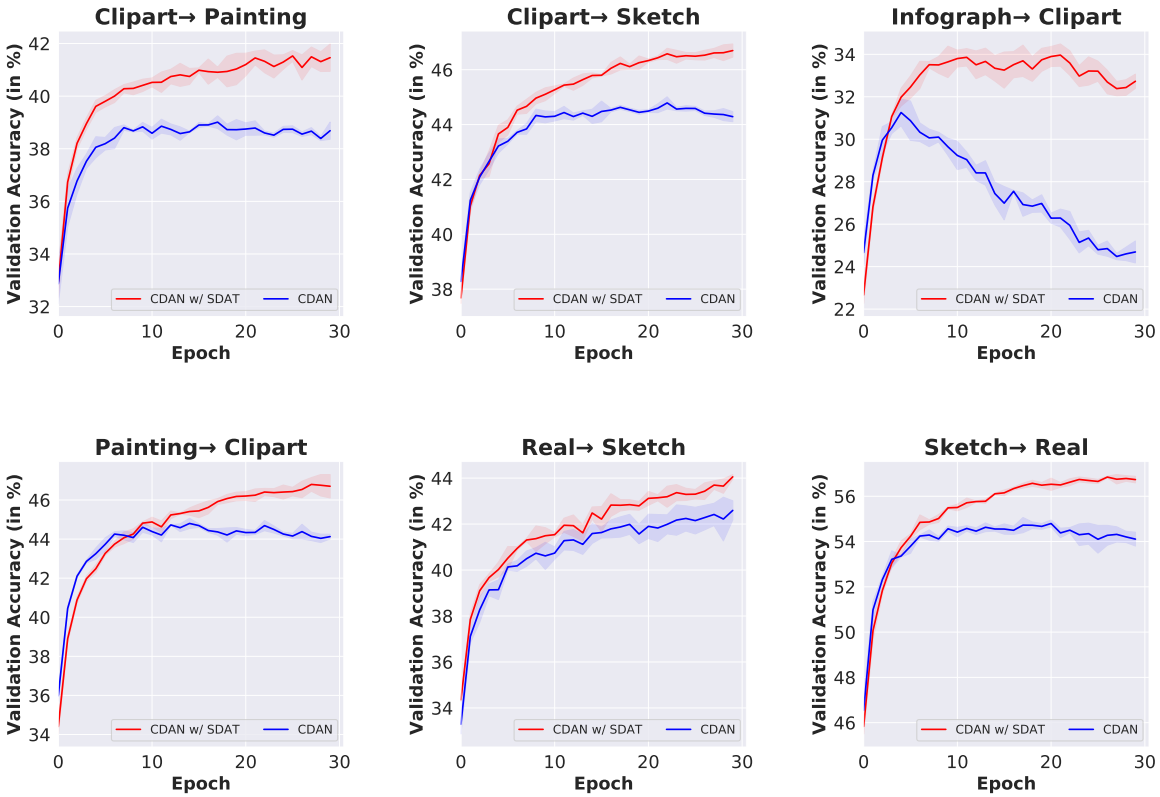


Figure 5. Validation Accuracy across epochs on different splits of DomainNet. We run on three different random seeds and plot the error bar indicating standard deviation across runs. CDAN w/ SDAT consistently outperforms CDAN across different splits of DomainNet.

experimental runs (18 for CDAN w/ SDAT (Our proposed method) and 18 for CDAN baseline). Due to the large computational complexity of each experiment (≈20 hrs each), we have presented results for multiple trials on a subset of splits. We find (in Table 14) that our method can outperform the baseline average in each of the 6 cases, establishing significant improvement across all splits. However, we found that due to the large size of DomainNet, the average increase (across three different trials) is close to the reported increase in all cases (Table 14), which also serves as evidence of the soundness of reported results (for remaining splits). We also present additional statistics below for establishing soundness.

If the proposed method is unstable, there is a large variance in the validation accuracy across epochs. For analyzing the stability of SDAT, we show the validation accuracy plots in Figure 5 on six different splits of DomainNet. We find that our proposed SDAT improves over baselines consistently across epochs without overlap in confidence intervals in later epochs. This also provides evidence for the authenticity and stability of our results. We also find that in some cases, like when using the Infographic domain as a source, our proposed SDAT also significantly *stabilizes the training* (Figure 5 *inf → clp*).

One of the other ways of reporting results reliably proposed by the concurrent work [3] (Section 4.4) involves reporting the median of accuracy across the last few checkpoints. The median is a measure of central tendency which ignores outlier results. We also report the median of validation accuracy for our method *across all splits* for the last five epochs. It is observed that we observe similar gains for median accuracy (in Table 15) as reported in Table 2.

As the Office-Home dataset is smaller (i.e., 44 images per class) in comparison to DomainNet we find that there exists some variance in baseline CDAN results (This is also reported in the well-known benchmark for DA [28]). For establishing the empirical soundness, we report results of 4 different dataset splits on 3 seeds. It can be seen in Table 16 that even though there is variance in baseline results, our combination of CDAN w/ SDAT can produce consistent improvement across different random seeds. This further establishes the empirical soundness of our procedure.

Table 14. DomainNet experiments over 3 different seeds (with ResNet backbone). We report the mean, standard deviation, reported increase and average increase in the accuracy (in %).

Split	CDAN	CDAN w/ SDAT	Reported Increase (Table 2)	Average Increase
clp→pnt	38.9 ± 0.1	41.5 ± 0.3	+2.6	+2.6
skt→rel	55.1 ± 0.2	57.1 ± 0.1	+2.2	+2.0
pnt→clp	44.5 ± 0.3	47.1 ± 0.3	+3.4	+2.6
rel→skt	42.4 ± 0.4	43.9 ± 0.1	+1.6	+1.5
clp→skt	44.9 ± 0.2	47.3 ± 0.1	+2.3	+2.4
inf→clp	31.4 ± 0.5	34.2 ± 0.3	+2.3	+2.7

Table 15. Median accuracy of last 5 epochs on DomainNet dataset with CDAN w/ SDAT. The number in the parenthesis indicates the increase in accuracy with respect to CDAN.

Target (→) Source (↓)	**clp**	**inf**	**pnt**	**real**	**skt**	**Avg**
clp	-	21.9 (+1.7)	41.6 (+3.0)	56.5 (+1.3)	46.4 (+2.0)	41.6 (+2.0)
inf	32.4 (+7.9)	-	29.8 (+7.0)	46.7 (+12.7)	25.6 (+5.4)	33.6 (+8.2)
pnt	47.2 (+2.9)	21.0 (+1.1)	-	57.6 (+1.0)	41.5 (+2.4)	41.8 (+1.8)
real	56.5 (+0.7)	25.5 (+0.9)	53.9 (+0.5)	-	43.5 (+1.3)	44.8 (+0.8)
skt	59.1 (+3.0)	22.1 (+1.7)	48.2 (+3.1)	56.6 (+2.9)	-	46.5 (+2.7)
Avg	48.8 (+3.6)	22.6 (+1.3)	43.4 (+3.4)	54.3 (+4.5)	39.2 (+2.8)	41.7 (+3.1)

Table 16. Office-Home experiments over 3 different seeds (with ResNet-50 backbone). We report the mean, standard deviation, reported increase and average increase in the accuracy (in %).

Split	CDAN	CDAN w/ SDAT	Reported Increase (Table **??**)	Average Increase
Ar→Cl	53.9 ± 0.2	55.5 ± 0.2	+1.7	+1.6
Ar→Pr	70.6 ± 0.4	72.1 ± 0.4	+1.6	+1.5
Rw→Cl	60.7 ± 0.5	61.8 ± 0.4	+1.3	+1.1
Pr→Cl	54.7 ± 0.4	55.5 ± 0.4	+0.6	+0.8

L. PyTorch Pseudocode for SDAT

In the code snippet below, we show that with a few changes in the code, SDAT can be easily integrated with any DAT algorithm. SDAT requires an additional forward pass and gradient computation, as shown below.

```
# task_loss_fn refers to the function to calculate task loss.
# (For classification settings, this can be Cross Entropy Loss).

# optimizer refers to the smooth optimizer which contains parameters of the feature extractor and classifier.
optimizer.zero_grad()
# ad_optimizer refers to standard SGD optimizer which contains parameters of domain classifier.
ad_optimizer.zero_grad()

# Calculate task loss
class_prediction, feature = model(x)
task_loss = task_loss_fn(class_prediction, label)
task_loss.backward()

```

```
# Calculate  ê (w)  and add it to the weights
optimizer.first_step()

# Calculate task loss and domain loss
class_prediction, feature = model(x)
task_loss = task_loss_fn(class_prediction, label)
domain_loss = domain_classifier(feature)
loss = task_loss + domain_loss
loss.backward()

# Update parameters (Sharpness-Aware update)
optimizer.second_step()
# Update parameters of domain classifier
ad_optimizer.step()
```

M. Adaptation for object detection

To further validate our approach's generality and extensibility, we did experiments on DA for object detection. We use the same setting as proposed in DA-Faster [8] with all domain adaptation components and use it as our baseline. We use the mean Average Precision at 0.5 IoU (mAP) as our evaluation metric. In object detection, the smoothness enhancement can be achieved in two ways (empirical comparison in Sec. M.2) :

a) DA-Faster w/ SDAT-Classification: Smoothness enhancement for classification loss.

b) DA-Faster w/ SDAT: Smoothness enhancement for the combined classification and regression loss.

M.1. Experimental Setup

We evaluate our proposed approach on object detection on two different domain shifts:

Pascal to Clipart ($P \rightarrow C$): Pascal [14] is a real-world image dataset which consists images with 20 different object categories. Clipart [24] is a graphical image dataset with complex backgrounds and has the same 20 categories as Pascal. We use ResNet-101 [21] backbone for Faster R-CNN [40] following [42].

Cityscapes to Foggy Cityscapes ($C \rightarrow Fc$): Cityscapes [10] is a street scene dataset for driving, whose images are collected in clear weather. Foggy Cityscapes [44] dataset is synthesized from Cityscapes for the foggy weather. We use ResNet-50 [21] as the backbone for Faster R-CNN for experiments on this task. Both domains have the same 8 object categories with instance labels.

M.2. Results

Table 17 shows the results on two domain shifts with varying batch size (*bs*) during training. We find that only smoothing w.r.t. classification loss is much more effective (SDAT-Classification) than smoothing w.r.t. combined classification and regression loss (SDAT). On average, SDAT-Classification produces an mAP gain of 2.0% compared to SDAT, and 2.8% compared to DA-Faster baseline.

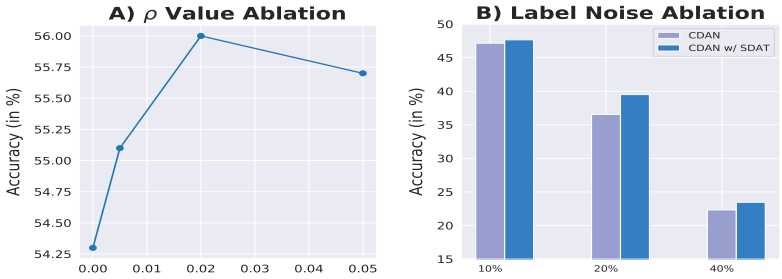
Table 17. Results on DA for object detection.

Method	$C \rightarrow Fc$ *(bs=2)*	$P \rightarrow C$ *(bs=2)*	$P \rightarrow C$ *(bs=8)*
DA-Faster [8]	35.21	29.96	26.40
DA-Faster w/ SDAT	37.47	29.04	27.64
DA-Faster w/ SDAT-Classification	**38.00**	**31.23**	**30.74**

The proposed SDAT-Classification significantly outperforms DA-Faster baseline and improves mAP by 1.3% on $P \rightarrow C$ and by 2.8% on $C \rightarrow Fc$. It is noteworthy that increase in performance of SDAT-Classification is consistent even after training with higher batch size ($bs = 8$) achieving improvement of 4.3% in mAP. Table 17 also shows that even DA-Faster w/ SDAT (i.e. smoothing both classification and regression) outperforms DA-Faster by 0.9 % on average. The improvement due to SDAT on adaptation for object detection shows the generality of SDAT across techniques that have some form of adversarial component present in the loss formulation.

N. Discussion

How much smoothing is optimal?: Figure 6A shows the ablation on ρ value (higher ρ value corresponds to more smoothing) on the Ar→Cl from Office-Home dataset with CDAN backbone. The performance of the different values of ρ is



C) Smoothing Loss Components

Smooth Task	Smooth Adv	Accuracy
✗	✗	54.3
✗	✓	51.0
✓	✗	**55.7**
✓	✓	54.9

Figure 6. Analysis of SDAT for Ar → Cl split of Office-Home dataset. **A)** Variation of target accuracy with maximum perturbation ρ. **B)** Comparison of accuracy of SDAT with DAT for different ratio of label noise. **C)** Comparison of accuracy when smoothing is applied to various loss components.

Table 18. Performance comparison across different loss smoothing techniques on Office-Home. SDAT (with ResNet-50 backbone) outperforms other smoothing techniques in each case consistently.

Method	Ar→Cl	Cl→Pr	Rw→Cl	Pr→Cl	Avg	
DAT	54.3	69.5	60.1	55.3	59.2	
VAT	54.6	70.7	60.8	54.4	60.1	(+0.9)
SWAD	54.6	71.0	60.9	55.2	60.4	(+1.2)
LS	53.6	71.6	59.9	53.4	59.6	(+0.4)
SAM	54.9	70.9	59.2	53.9	59.7	(+0.5)
SDAT	**56.0**	**73.2**	**61.4**	**55.9**	**61.6**	(+2.4)

higher than the baseline with $\rho = 0$. It can be seen that $\rho = 0.02$ works best among all the different values and outperforms the baseline by at least 1.5%. We found that the same ρ value usually worked well across domains in a dataset, but different ρ was optimal for different datasets. More details about optimum ρ is in App. I.

Which components benefit from smooth optima?: Fig. 6C shows the effect of introducing smoothness enhancement for different components in DAT. For this we use SAM on a) task loss (SDAT) b) adversarial loss (SDAT w/ adv) c) both task and adversarial loss (SDAT-all). It can be seen that smoothing the adversarial loss component (SDAT w/ adv) reduces the performance to 51.0%, which is significantly lower than even the DAT baseline.

Is it Robust to Label Noise?: In practical, real-world scenarios, the labeled datasets are often corrupted with some amount of label noise. Due to this, performing domain adaptation with such data is challenging. We find that smoother minima through SDAT lead to robust models which generalize well on the target domain. Figure 6B provides the comparison of SGD vs. SDAT for different percentages of label noise injected into training data.

Is it better than other smoothing techniques? To answer this question, we compare SDAT with different smoothing techniques originally proposed for ERM. We specifically compare our method against DAT, Label Smoothing (LS) [46], SAM [15] and VAT [36]. [45] recently showed that these techniques produce a significantly smooth loss landscape in comparison to SGD. We also compare with a very recent SWAD [6] technique which is shown effective for domain generalization. For this, we run our experiments on four different splits of the Office-Home dataset and summarize our results in Table 18. We find that techniques for ERM (LS, SAM and VAT) fail to provide significant consistent gain in performance which also confirms the requirement of specific smoothing strategies for DAT. We find that SDAT even outperforms SWAD on average by a significant margin of 1.2%. Additional details regarding the specific methods are provided in App. H.

Does it generalize well to other DA methods?: We show results highlighting the effect of smoothness (SDAT) on DANN [17] and GVB-GD [11] in Table 19 with ResNet-50 and ViT backbone. DANN w/ SDAT leads to gain in accuracy on both DomainNet and Office-Home dataset. We observe a significant increase (average of +3.3%) with DANN w/ SDAT (ViT backbone) on Office-Home dataset. SDAT leads to a decent gain in accuracy on Office-Home dataset with GVB-GD despite the fact that GVB-GD is a much stronger baseline than DANN. We primarily focused on CDAN and CDAN + MCC for the main results as we wanted to establish that SDAT can improve on even SOTA DAT methods for showing its effectiveness. Overall, we have shown results with four DA methods (CDAN, CDAN+MCC, DANN, GVB-GD) and this shows that SDAT is a generic method that can applied on top of any domain adversarial training based method to get better performance.

Table 19. Analysing the effect of SDAT on DANN (on DomainNet and Office-Home with ResNet-50 and ViT-B/16 respectively) and GVB-GD (on Office-Home with ResNet-50).

DomainNet		**clp→pnt**	**skt→pnt**	**inf→real**	**skt→clp**
DANN	RN-50	37.5	43.9	37.7	53.8
DANN w/ SDAT		38.9 (+1.4)	45.7 (+1.8)	39.6 (+1.9)	56.3 (+2.5)
Office-Home		**Ar→Cl**	**Cl→Pr**	**Rw→Cl**	**Pr→Cl**
GVB-GD	RN-50	56.4	74.2	59.0	55.9
GVB-GD w/ SDAT		57.6 (+1.2)	75.4 (+1.2)	60.0 (+1.0)	56.6 (+0.7)
DANN		52.6	65.4	60.4	52.3
DANN w/ SDAT		53.4 (+0.8)	66.4 (+1.0)	61.3 (+0.9)	53.8 (+1.5)
DANN	ViT	62.7	81.8	68.5	66.5
DANN w/ SDAT		68.0 (+5.3)	82.4 (+0.6)	73.4 (+4.9)	68.8 (+2.3)