ICML | 2022
Thirty-ninth International Conference on Machine Learning

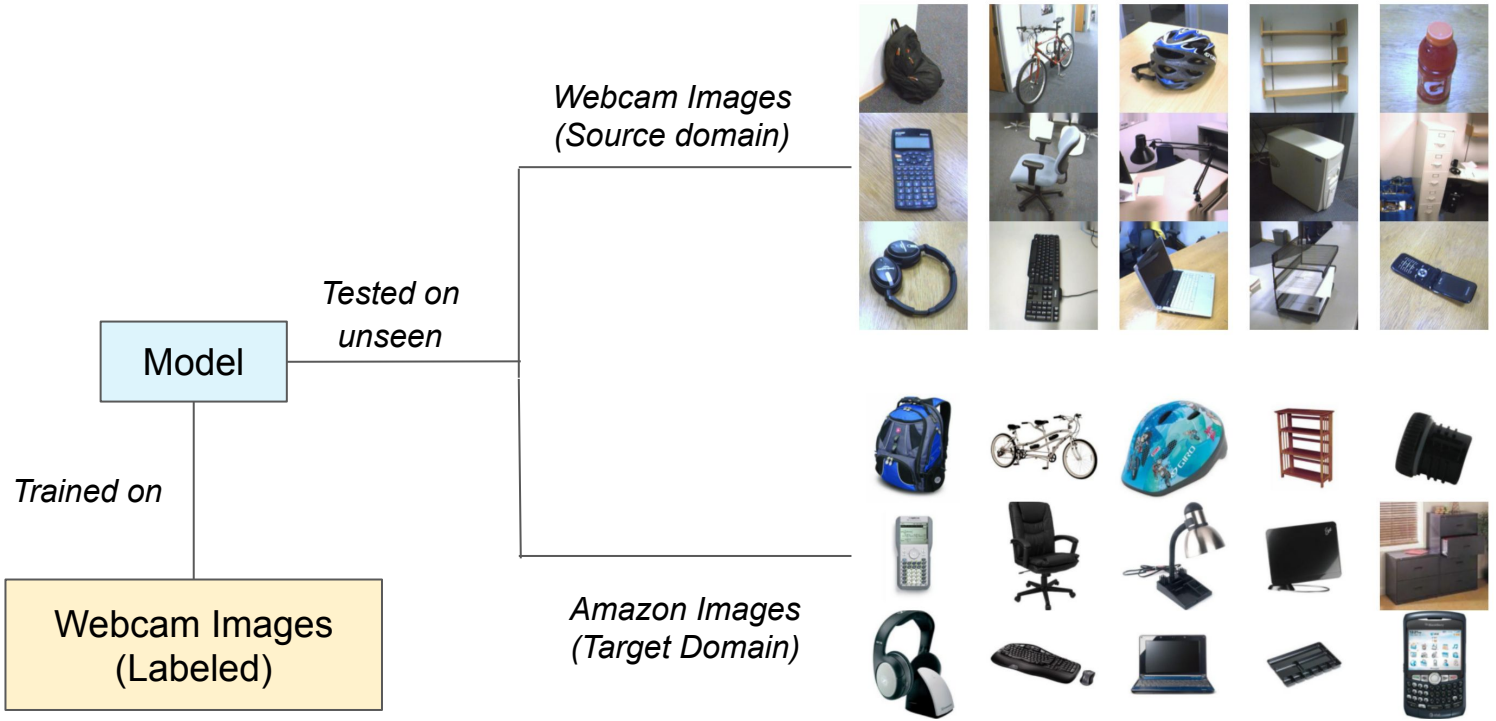
A Closer Look at Smoothness in Domain Adversarial Training

Harsh Rangwani*, Sumukh K Aithal*, Mayank Mishra, Arihant Jain,
R. Venkatesh Babu

Indian Institute of Science, Bangalore

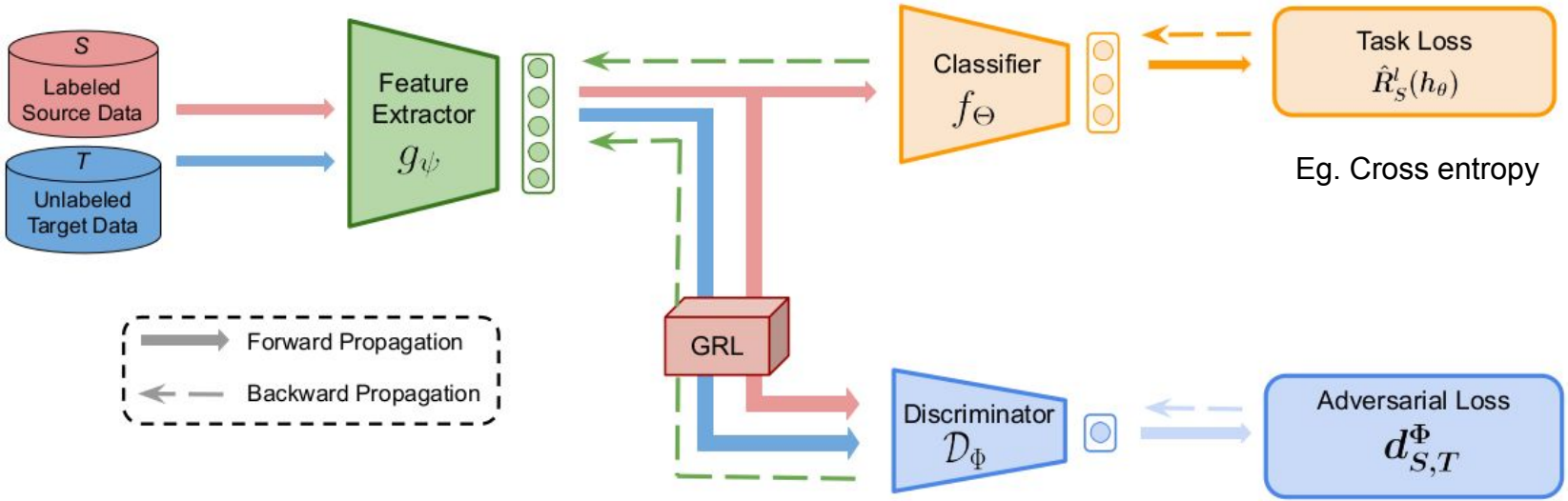
* indicates Equal Contribution

Domain Adaptation



Images taken from Office-31 dataset.

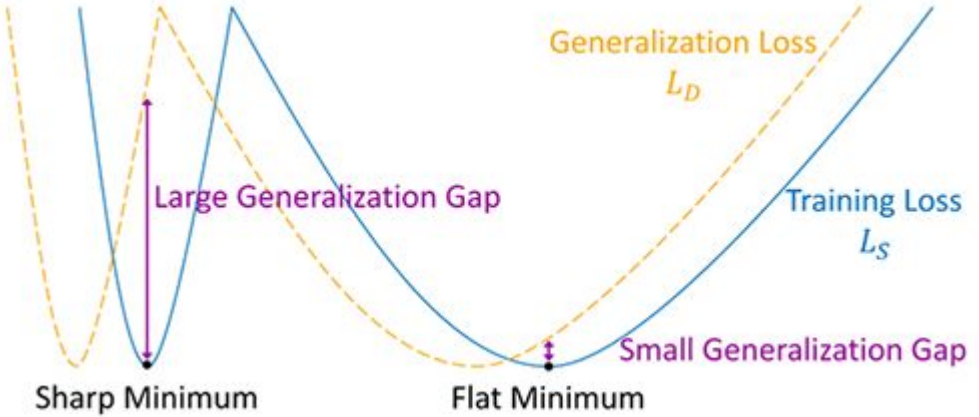
Domain Adversarial Training (DAT)



Overall Objective: **Task Loss + Adversarial Loss**

Sharp vs Flat Minima

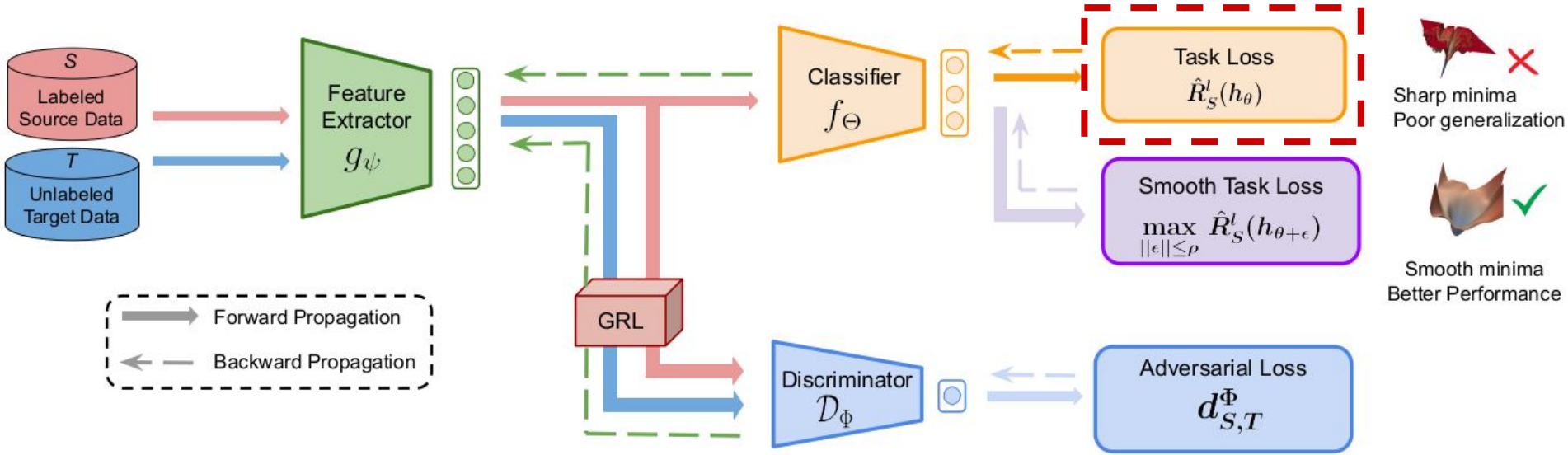
Converging to a flatter (smooth) region in loss landscape with supervised learning leads to effective **generalization on same domain** as compared to sharp minima.



We analyze the effect of smoothness (i.e. flatness) enhancement for **Domain Adversarial Training**.

Figure Credit: (Link)

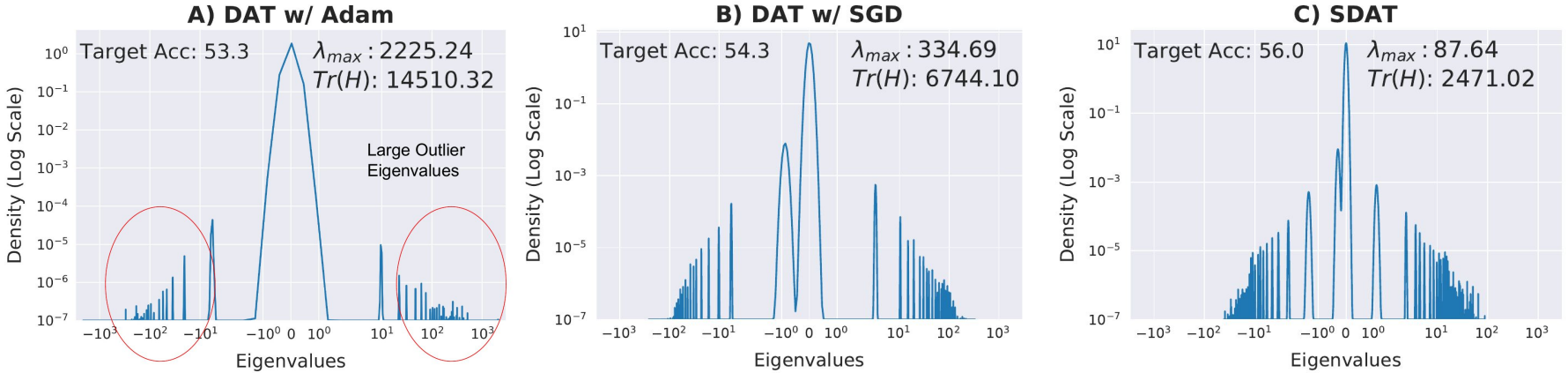
Analysis of Smoothness of Task Loss



$$R^l_S(h_\theta) = \mathbb{E}_{x\sim P_S}\left[l(h_\theta(x), y(x))\right]$$

Smoothness of Task Loss Is Beneficial

- Hessian of Source Risk (Task Loss) is used to analyze the loss landscape.
- Plot of spectral density of eigen values of Hessian is given below.

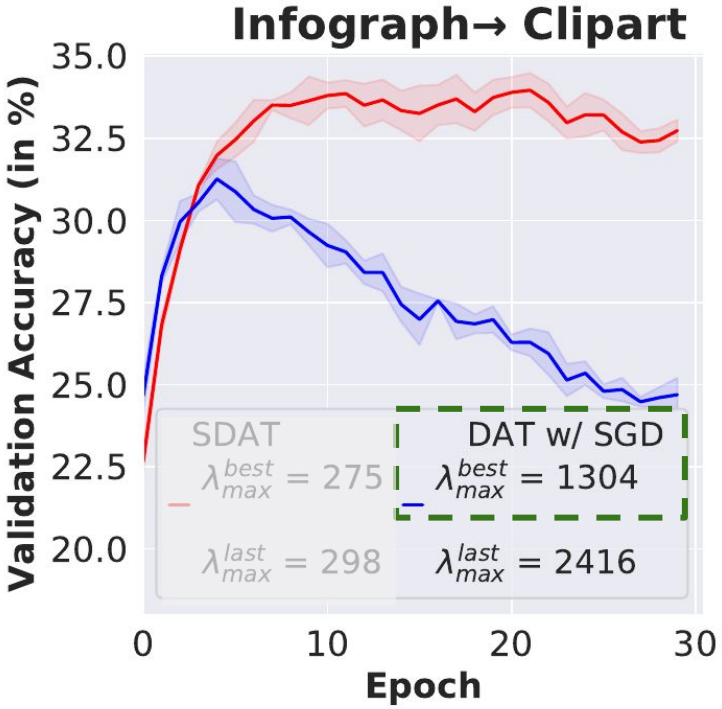


λ_{max} : maximum eigenvalue of the Hessian Tr(H): Trace of the Hessian

Low λ_{max} and low Tr(H) indicate convergence to smooth region.

Smoothness stabilizes Domain Adversarial Training

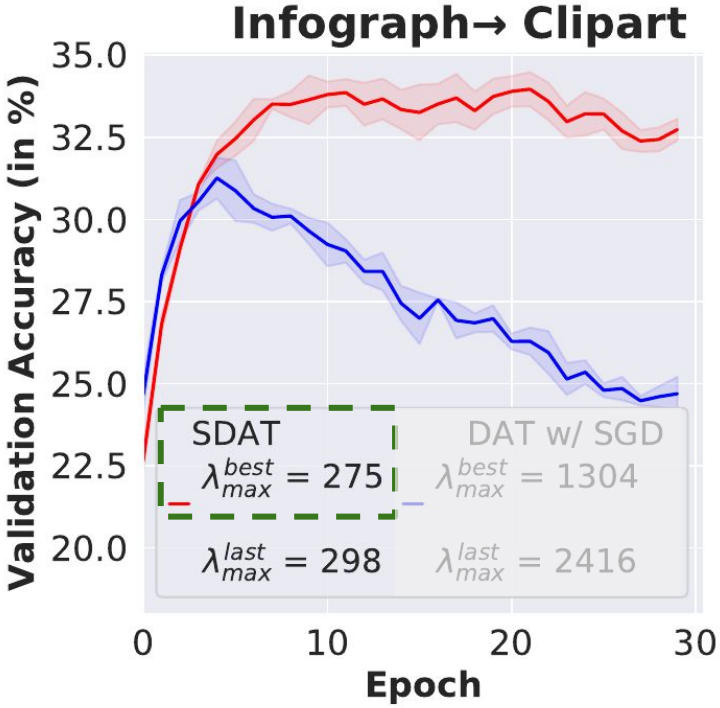
- As λ_{max} increases (decrease in smoothness of landscape), the training becomes unstable for SGD leading to a drop in validation accuracy.



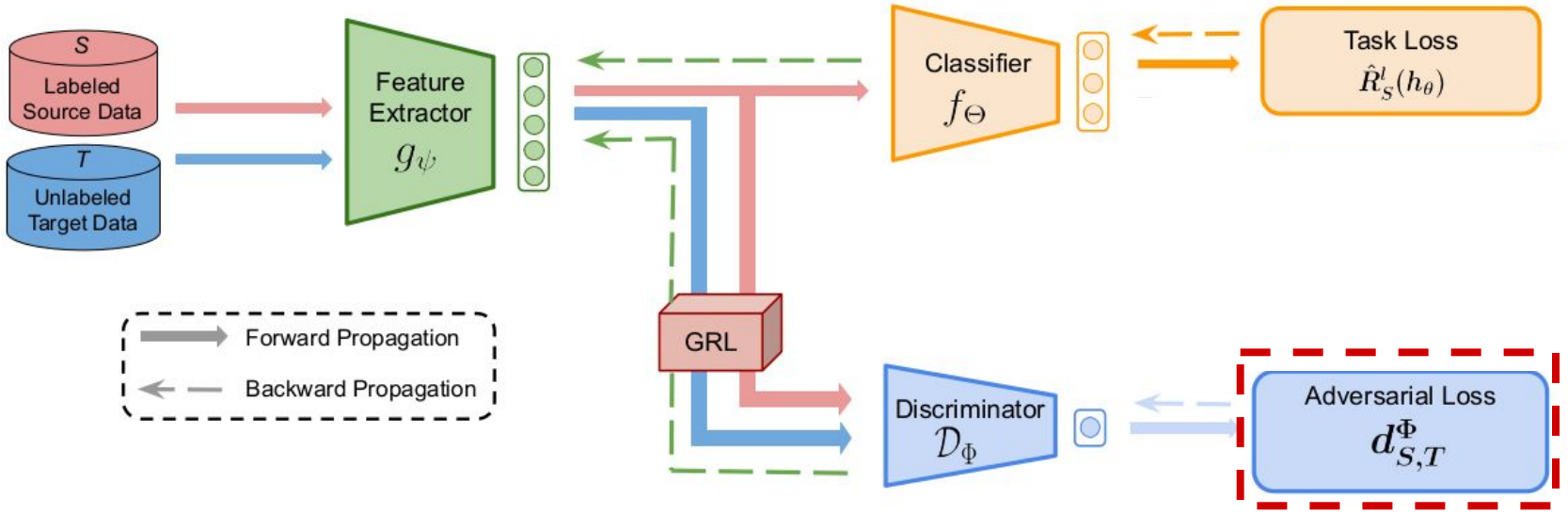
Smoothness stabilizes Domain Adversarial Training

- As λ_{max} increases (decrease in smoothness of landscape), the training becomes unstable for SGD leading to a drop in validation accuracy.

- In the smooth DAT, the λ_{max} remains low across epochs, leading to **stable** and better validation accuracy curve.



Analysis of Smoothness of Adversarial Loss



$$d_{S,T}^{\Phi} = \mathbb{E}_{x\sim P_S}\left[\log\left(\mathcal{D}_\Phi\left(g_\psi(x)\right)\right)\right] + \mathbb{E}_{x\sim P_T}\log\left[1-\mathcal{D}_\Phi\left(g_\psi(x)\right)\right.$$

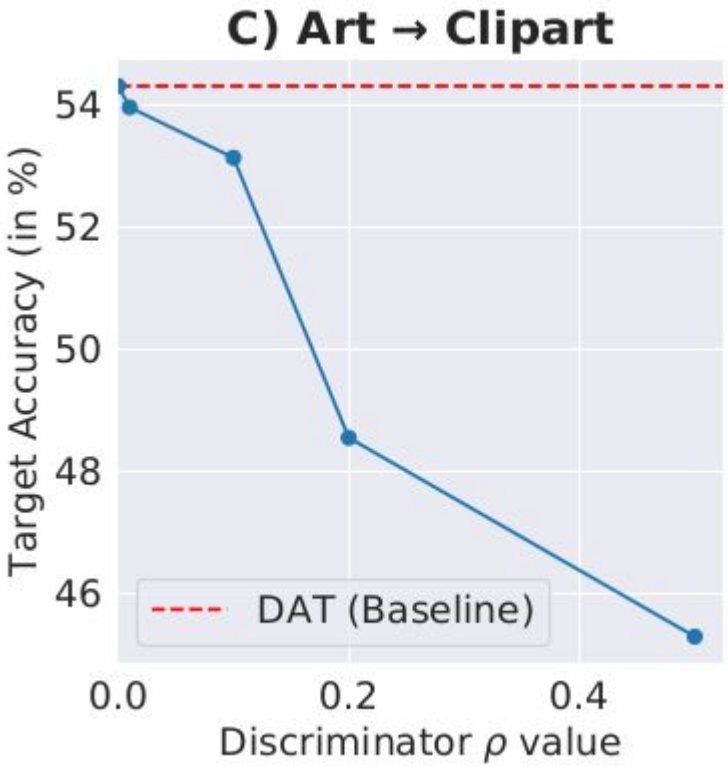
Analysis of Smoothness of Adversarial Loss

Theorem 2 (Informal Remark): We show that for a class of functions (gradient lipschitz) that the discriminator is suboptimal between source and target domain, when smooth version of adversarial loss is used.

Analysis of Smoothness of Adversarial Loss

Theorem 2 (Informal Remark): We show that for a class of functions (gradient lipschitz) that the discriminator is suboptimal between source and target domain, when smooth version of adversarial loss is used.

Empirical: As the smoothness increases (ρ), the target accuracy decreases indicating that smoothing adversarial loss leads to suboptimal generalization.



Increasing Smoothness →

Smooth Domain Adversarial Training (SDAT)

Sharpness-Aware Minimization (SAM) [1]

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

$$\hat{\epsilon}(\theta) \approx \arg\max_{||\epsilon|| \leq \rho} L_{obj}(\theta) + \epsilon^T \nabla_\theta L_{obj}(\theta)$$

$$= \rho \nabla_\theta L_{obj}(\theta) / ||\nabla_\theta L_{obj}(\theta)||_2$$

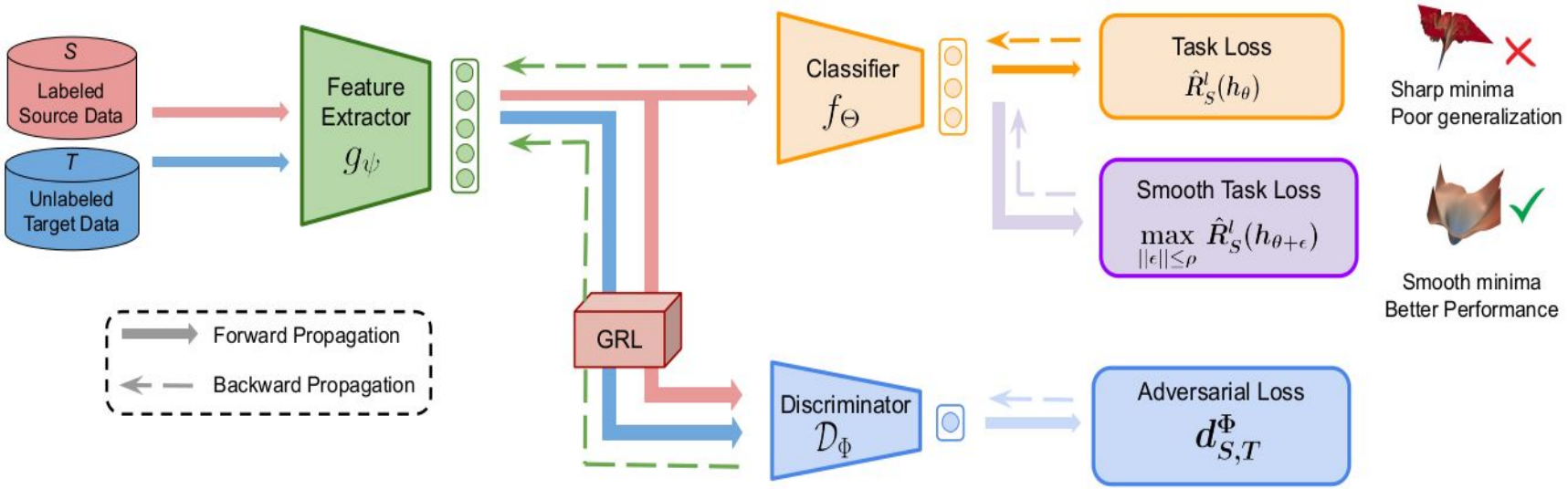
Smooth Domain Adversarial Training:

Smooth Minima w.r.t task loss (Empirical Source Risk)

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

[1] Foret, Pierre, et al. "Sharpness-aware minimization for efficiently improving generalization.", ICLR 2021.

Smooth Domain Adversarial Training (SDAT)



SDAT: Easy to Implement and Integrate

- SDAT can be easily integrated with various Domain Adversarial Training methods with only a few lines of changes in the code.

- SDAT leads to significant gain in the accuracy on the target domain on combination with domain adversarial methods.

```
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()
```

SDAT in Practice (Office-Home Dataset)

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 (He et al., 2016)	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 (Ganin et al., 2016)		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* (Long et al., 2018)		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 (Zhang et al., 2019)		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 (Acuna et al., 2021)		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 (Tang et al., 2020)		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**

SDAT in Practice (Office-Home Dataset)

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 (He et al., 2016)	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 (Ganin et al., 2016)		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* (Long et al., 2018)		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 (Zhang et al., 2019)		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 (Acuna et al., 2021)		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 (Tang et al., 2020)		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 (Yang et al., 2021)	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**

SDAT in Practice (Office-Home Dataset)

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 (He et al., 2016)	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 (Ganin et al., 2016)		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* (Long et al., 2018)		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 (Zhang et al., 2019)		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 (Acuna et al., 2021)		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 (Tang et al., 2020)		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 (Yang et al., 2021)	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**

Performance on Large Scale DomainNet

Results are shown with CDAN w/ SDAT.

The number in the parenthesis refers to the increase in accuracy with respect to 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)

SDAT improves over SOTA i.e. TVT [1] and CDTrans [2]

	TVT	CDTrans	SDAT
Need of additional modules/networks	✓	✓	✗
Memory requirement (For training)	~35 GB	~26.3 GB	<12 GB
Pretraining	ImageNet-21k	ImageNet-1k	ImageNet-1k
Accuracy (Office-Home)	83.6%	80.5%	**84.3%**
Accuracy (VisDA-17)	83.2%	88.4%	**89.8%**

[1] Yang, J., Liu, J., Xu, N., & Huang, J. (2021). TVT: Transferable Vision Transformer for Unsupervised Domain Adaptation. arXiv. https://doi.org/10.48550/ARXIV.2108.05988

[2] Xu, T., Chen, W., Wang, P., Wang, F., Li, H., & Jin, R. (2021). Cdtrans: Cross-domain transformer for unsupervised domain adaptation.ICLR 2022.

Conclusion

TLDR: Smooth Minima with respect to task loss leads to effective generalization on the target domain.

→ SDAT is effective across tasks and benchmarks.

→ Can be combined easily with any of the domain adversarial methods.

→ Very easy to integrate in any framework with few lines of code.

→ Provides consistent improvement across both Convnets and Vision-Transformer based Architecture.

Thank You

Code: https://github.com/val-iisc/sdat
Paper: https://arxiv.org/abs/2206.08213