Unsupervised Domain Adaptation

Final Project Presentation

Michele Yin, Roberto Mazzaro, Andrea Bonora, Filippo Daniotti, Giovanni Ambrosi

Outline

1. Introduction
2. Works on DANN
   - DANN + Discrepancy loss
   - DANN + improvement for Adversarial methods and combinations
   - Incremental DANN
3. Gradual self training
   - Comparison with other datasets
   - Ablation study
4. Conclusions
   - Future works
   - Our opinion

Intro - Domain Adaptation

Labelled dataset: when we train networks

- **Pros:** training results in good performances
- **Cons:** expensive -> we can't have a label dataset for each application

Unlabelled data: usually for real applications we only have unlabeled data

- **Pros:** Almost free -> we want to exploit them
- **Cons:** domain shift

$\{X_s, Y_s\}$

$\{X_t\}$

Intro - Our work

Previously:

- General introduction to **Domain Adaptation**
- More specifically on **UDA** for **classification**
- Overview of different UDA techniques
- Test of some standard methods

Today:

- Work over DANN: **improve** DANN with **other methods and combination** of them
- Gradual self-training: in **depth ablation analysis** of a single method
- All our work can be found here https://github.com/filippodaniotti/TACV-DA-project

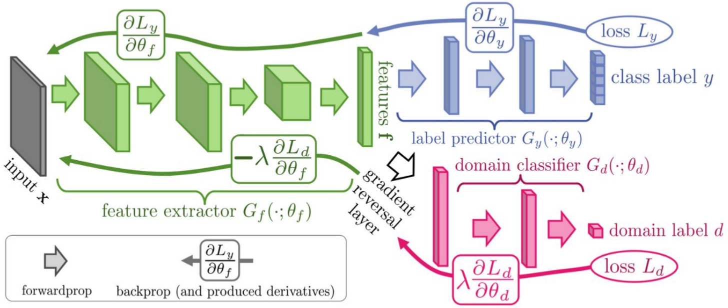
Experiment Setting

Ideas:

- Start with **DANN**
- **Improve** it with newer methods
- Try to **combine** more than one method together

Why DANN:

- **Simple** -> **fast** to implement and test
- Investigate if the concept make sense
- Don't look for the greatest accuracy



Dataset: Office31

- **Pros:** more complex than digits, lighter than OfficeHome
- **Cons:** unbalanced in the 2 domains

Office31

WEBCAM (795 images)

AMAZON (2817 images)

DANN + Discrepancy loss

- Starting from DANN
  - Add a loss at the output of the feature extractor
    - Simple and effective
  - Align the features before classification and discrimination
    - Maximum Mean Discrepancy Loss

$$\begin{aligned} MMD^2(P,Q) &= ||\mu_P - \mu_Q||^2_{\mathcal{F}} \\ &= \mathbb{E}_{\mathcal{X}\sim P}\left[k(x,x')\right] + \mathbb{E}_{\mathcal{Y}\sim Q}\left[k(y,y')\right] - 2\mathbb{E}_{\mathcal{X},\mathcal{Y}\sim P,Q}\left[k(x,y)\right] \end{aligned}$$

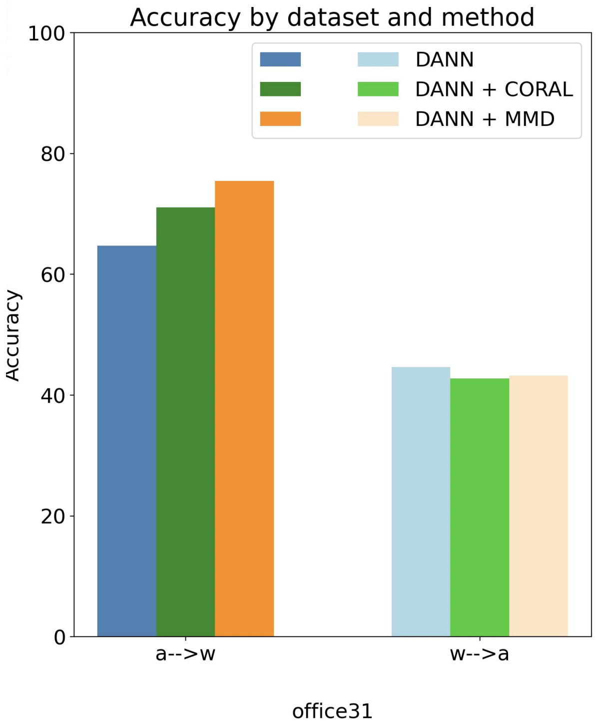
    - Coral Loss

$$\ell_{CORAL} = \frac{1}{4d^2}\|C_S - C_T\|^2_F$$

Obtained Results

- **MMD loss best on Amazon -> Webcam domain, gain over 10%**
- Coral loss improves of 6%
- Both losses don't provide improvements on Webcam -> Amazon direction, **unbalanced dataset**

Model	A->W A	A->W W	Gain	W->A W	W->A W	Gain
DANN	86.35	64.78	-	93.71	44.68	-
DANN + MMD	83.16	75.47	**+10.69**	95.60	42.02	-2.66
DANN + Coral	87.06	71.07	+6.29	94.34	43.25	-1.43



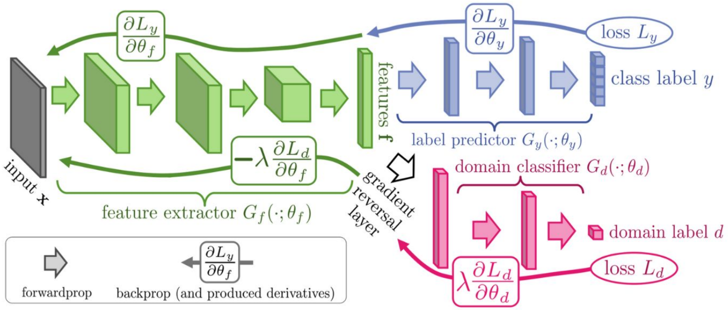
Experiment Setting

Ideas:

- Start with **DANN**
- **Improve** it with newer methods
- Try to **combine** more than one method together

Why DANN:

- **Simple** -> **fast** to implement and test
- Investigate if the concept make sense
- Don't look for the greatest accuracy

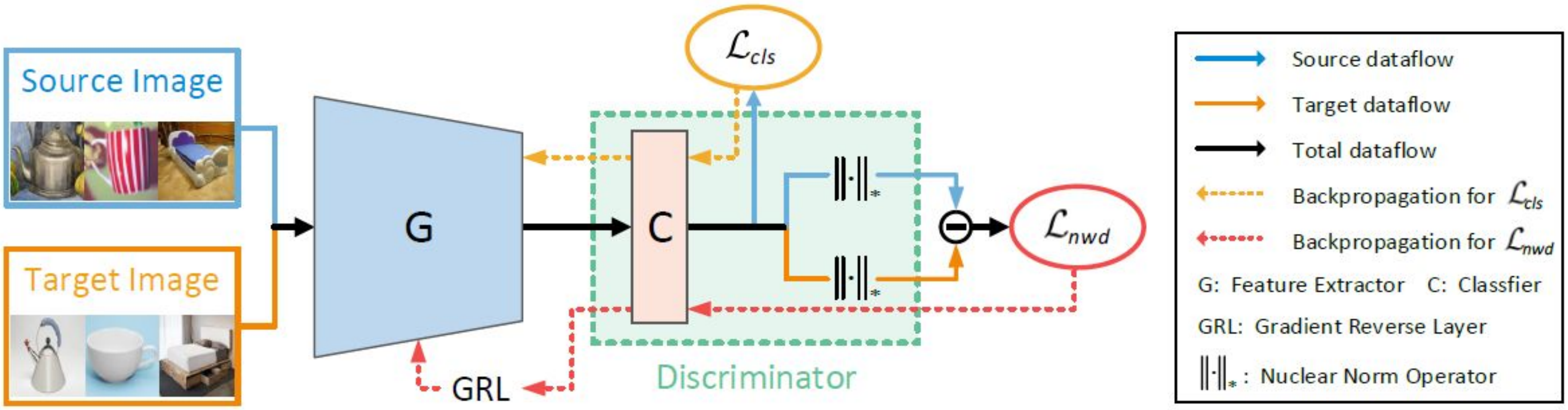


Dataset: Office31

- **Pros:** more complex than digits, lighter than OfficeHome
- **Cons:** unbalanced in the 2 domains

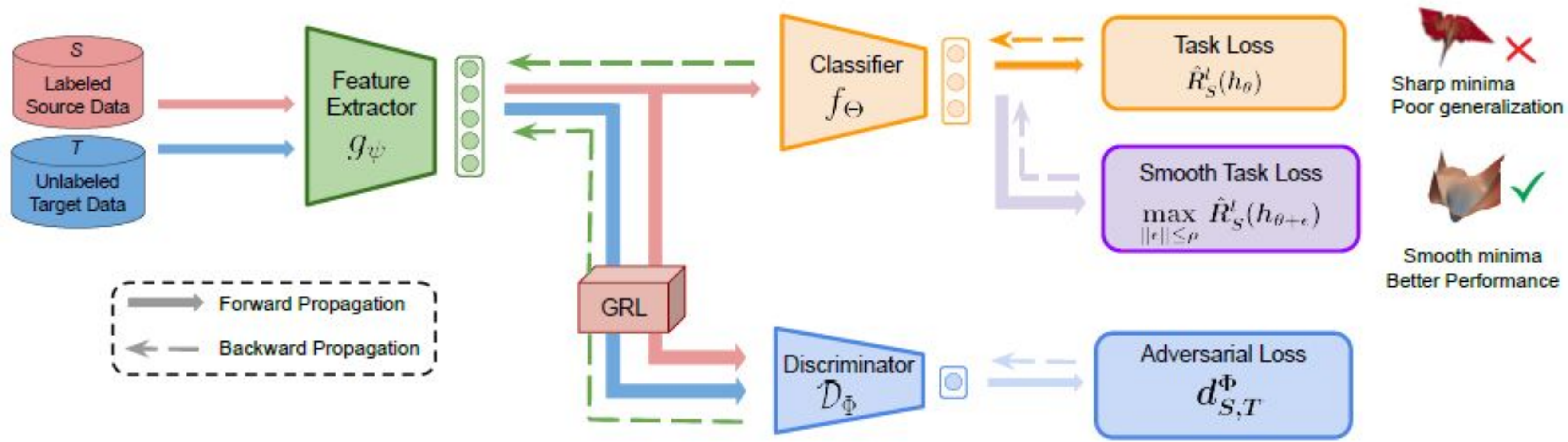
Methods Recap - DALN

- **Remove the discriminator**
- Use classifier + NWD module to discriminate the domain



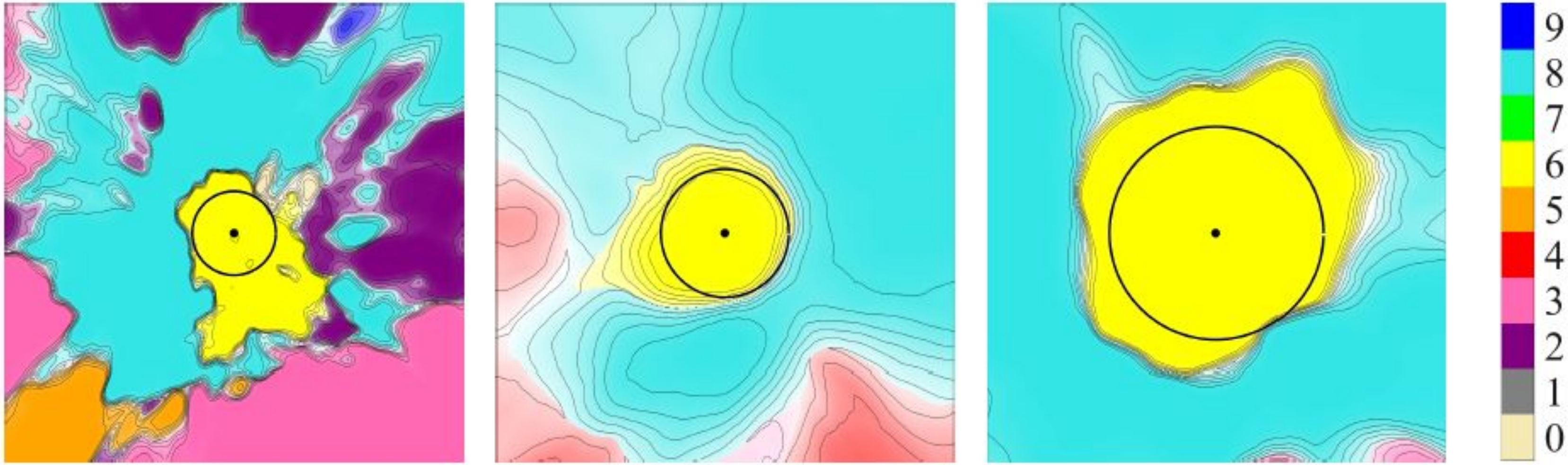
Methods Recap - SDAT

- Find **smoother minima** for classification loss
- **New optimizer** with additional gradient computation steps



Methods Recap - JREG

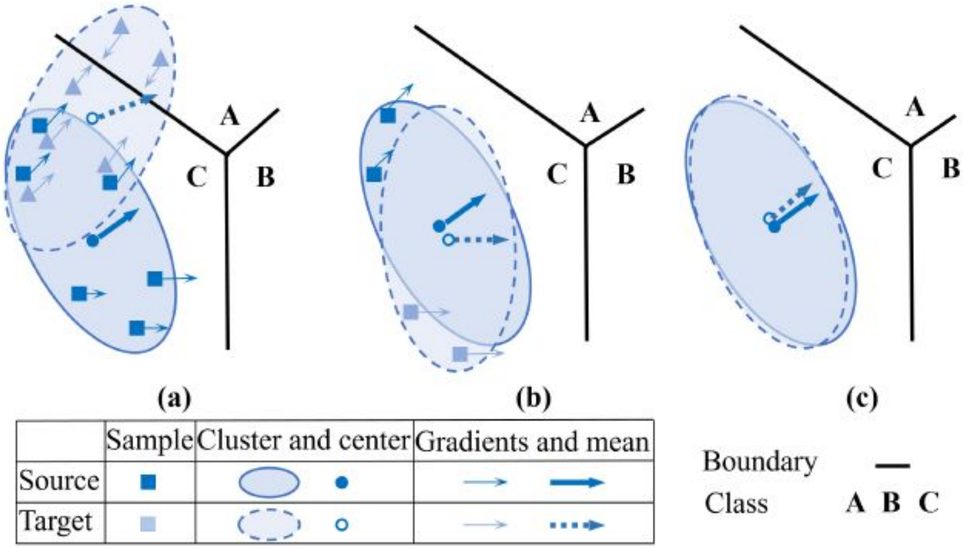
- Regularization method
- Push decision boundaries further away
- Used inside FGDA



(a) Without regularization (b) With L^2 regularization (c) With Jacobian regularization

Methods Recap - FGDA

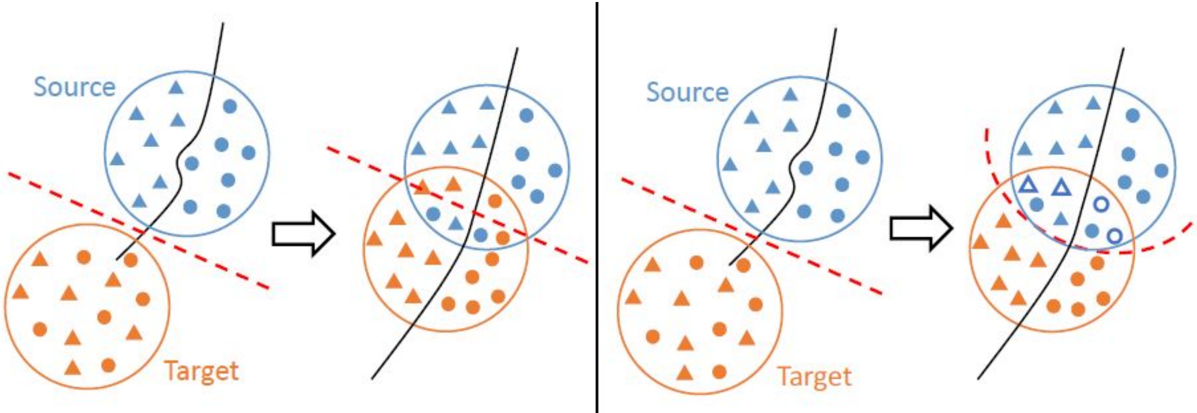
- **Constrain** feature **gradients** of two domains to have **similar distributions**
- Pseudo labels computed to obtain target loss
- Jacobian Regularization used inside



Methods Recap - RADA

- **Relabel** well aligned target **samples** as source domain
- Well aligned samples -> domain discriminator entropy higher than a threshold
- **Mixup** at feature level used with relabel samples to softly mix features
- Domain relabeling doesn't influence classification

- **No official implementation available**



(a) Previous Methods (b) Our Method

Combining methods - How

FGDA + DALN: no conflicts and lighter model

- FGDA use an additional grad_discriminator to align gradient distributions
- Adversarial discriminator can be substituted by DALN

RADA + FGDA: no conflicts

- RADA change domain labels but doesn't influence classification task
- Just add FGDA

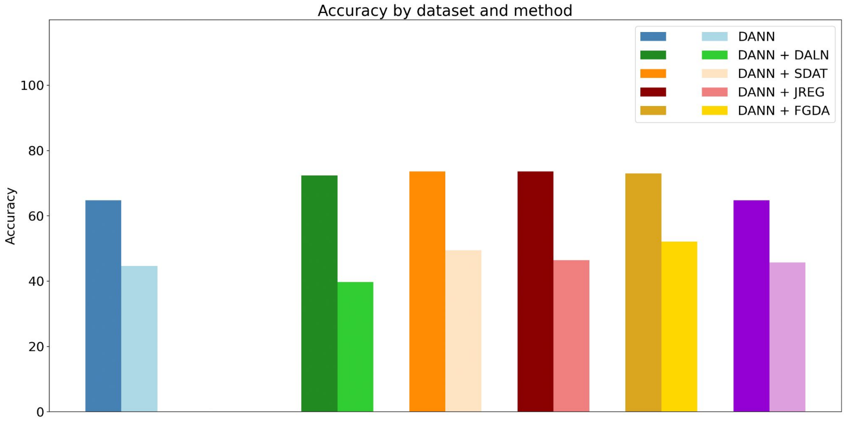
Any + SDAT: SDAT is a different optimizer so can be applied to any method

RADA + DALN: creates conflicts

- RADA use domain discriminator entropy as policy to re-align samples

Obtained Results

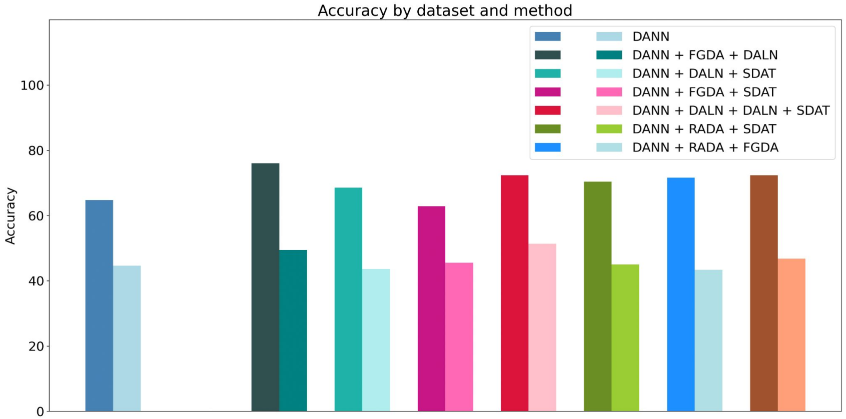
- Almost all methods improve DANN
- RADA in A->W test doesn't improve
  - Neither worsen and relabeling started at epoch 17
  - No official code and training parameters available
- DALN in W->A is suffering the dataset imbalance
- JREG very effective but FGDA improve it a lot in W->A



Model	A->W A	A->W W	Gain	W->A W	W->A W	Gain
DANN	86.35	64.78	-	93.71	44.68	-
DANN + DALN	83.16	72.33	+7.55	95.60	39.72	-4.96
DANN + SDAT	87.06	73.58	+8.80	94.34	49.47	+4.79
DANN + JREG	85.82	73.58	+8.80	94.97	46.45	+1.77
DANN + FGDA	86.35	72.96	+8.18	96.86	52.13	**+7.45**
DANN + RADA	85.99	64.78	0	93.08	45.74	+1.06

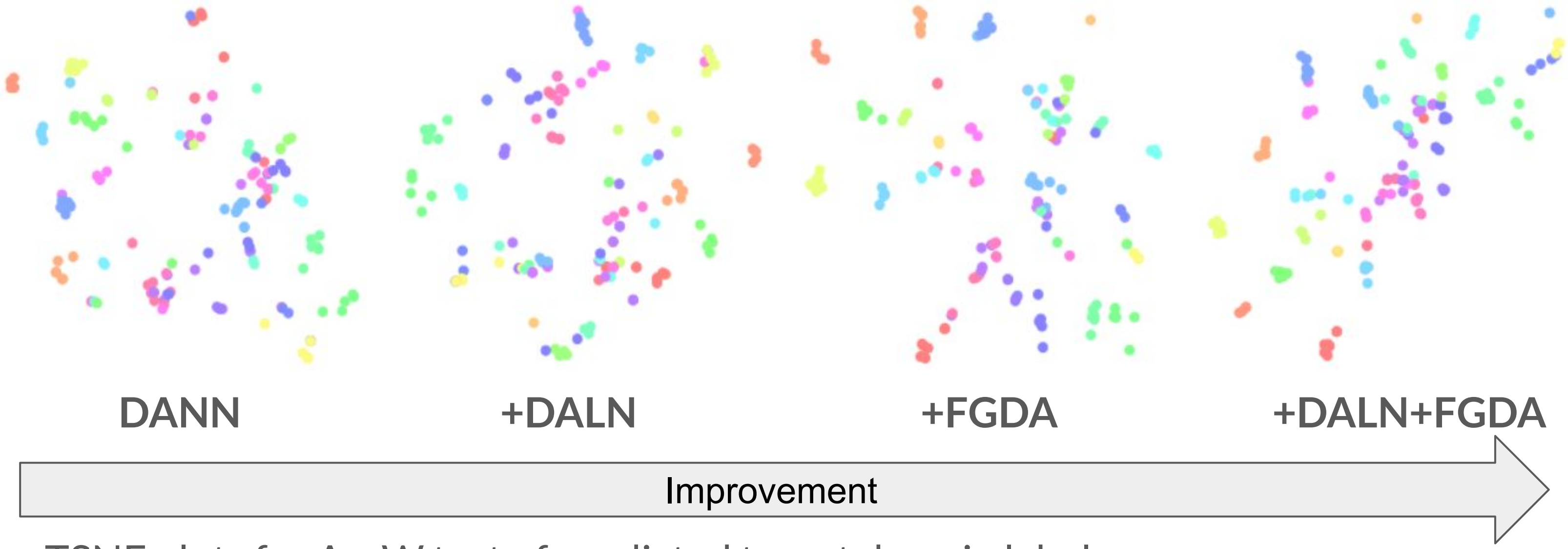
Obtained Results

- FGDA + DALN seems a good idea
  - Best method in A->W test with gain of +11.32
  - In W->A test suffer the poor performances of DALN in this direction
- RADA + FGDA might be a good idea
  - Increase RADA performances
  - Problem are RADA poor performances due to non optimal training params.
- SDAT very sensitive to training params. -> if not well selected decrease performances



Model	A->W A	A->W W	Gain	W->A W	W->A W	Gain
DANN + FGDA + DALN	84.75	**76.10**	+11.32	93.08	49.47	+4.79
DANN + DALN + SDAT	85.64	68.55	+3.77	91.82	43.62	-1.06
DANN + FGDA + SDAT	81.21	62.89	-1.89	93.71	45.57	+0.89
DANN + DALN + DALN + SDAT	85.82	72.33	+7.55	94.97	51.42	+6.74
DANN + RADA + SDAT	83.87	70.44	+5.66	94.34	45.04	+0.36
DANN + RADA + FGDA	83.87	71.70	+6.92	94.34	43.44	-1.24
DANN + RADA + FGDA + SDAT	84.57	72.33	+7.55	93.71	46.81	+2.13

TSNE Analisy

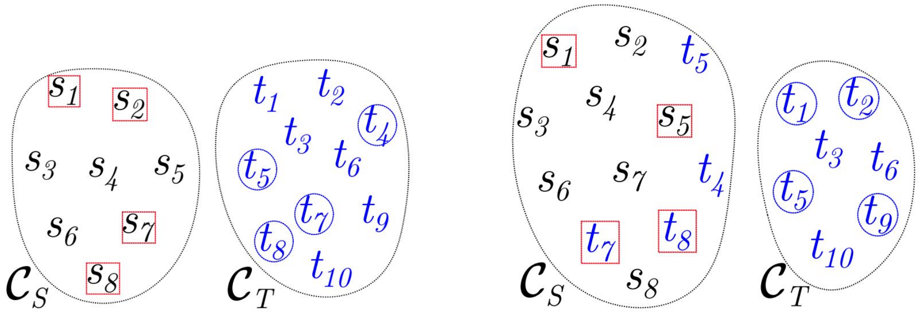


- TSNE plots for A->W test of predicted target domain labels
- Better inter class separation
- Better intra class compactness

Incremental Method

Idea:

- Start from a **trained model**
- Assign a pseudo label to ***k* samples**
- At each iteration train the model
  - **First**, train the model as usual
  - **Next**, only on the new pseudo labeled samples
- At the end, a model from scratch **only** with the target data

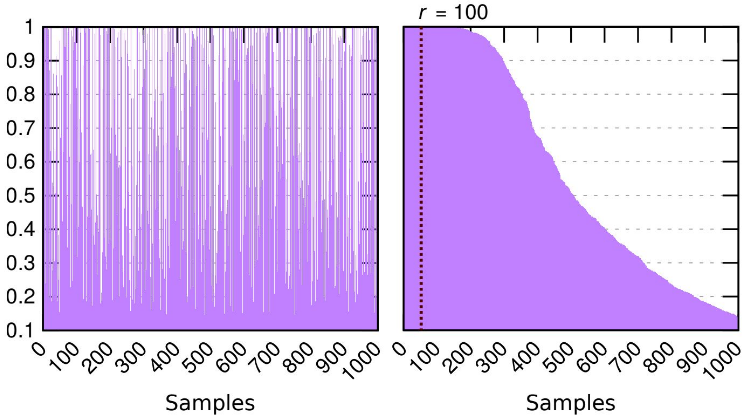


(a) 1^{st} iteration (b) 2^{nd} iteration

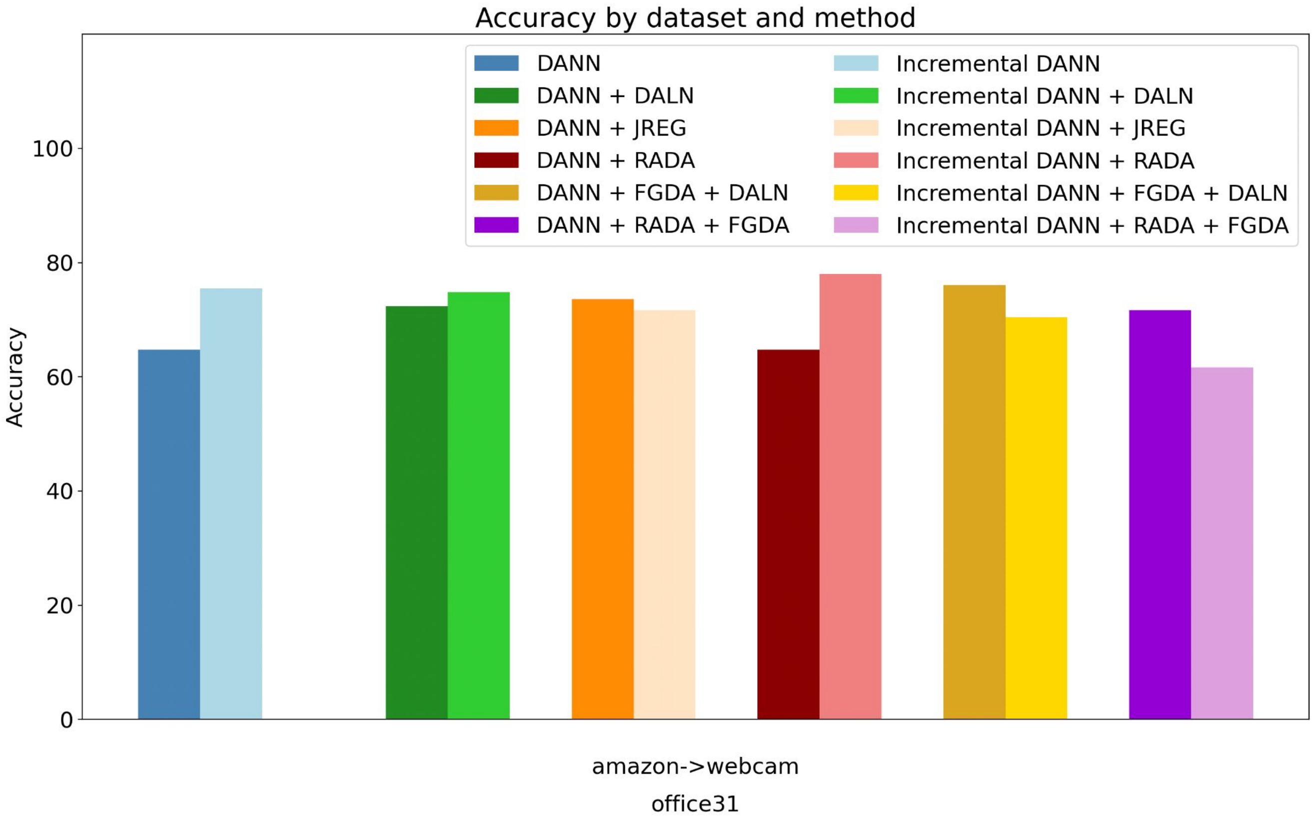
Incremental Method

How to assign a label to the data?

- **Confidence policy**: select the samples with the **highest confidence** in the classifier predictions
- **Possible issue:** samples with a very low confidence will distort the training of the model
- **Possible solution:** when the confidence is lower than a certain threshold assign all the remaining data to a label without training the model anymore



Results



Accuracy by dataset and method
DANN
DANN + DALN
DANN + JREG
DANN + RADA
DANN + FGDA + DALN
DANN + RADA + FGDA
Incremental DANN
Incremental DANN + DALN
Incremental DANN + JREG
Incremental DANN + RADA
Incremental DANN + FGDA + DALN
Incremental DANN + RADA + FGDA
Accuracy
0
20
40
60
80
100
amazon->webcam
office31

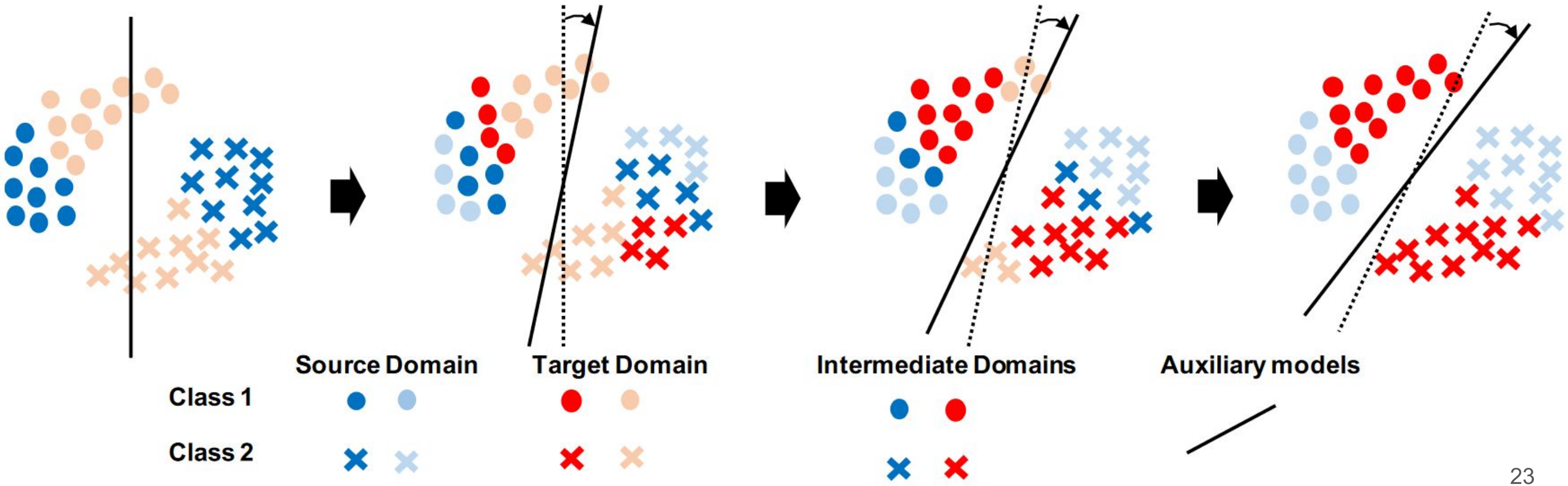
Adversarial Future Works

- Try different training parameters to possibly obtain better results in particular with RADA and SDAT
- Try different starting architectures (e.g. CDAN)
- Test on different datasets (e.g. MNIST or OfficeHome) to have a better understanding
- Try a different alignment measure for RADA not based on discriminator output allowing to fuse RADA with DALN
- Try different policies for selecting samples in the incremental method (e.g. k-NN)
- Test the incremental method with different hyperparameters setting

AuxSelfTrain

Key Idea:

- gradually replace source samples with target samples
- assign pseudo-labels through self-training



Experiments

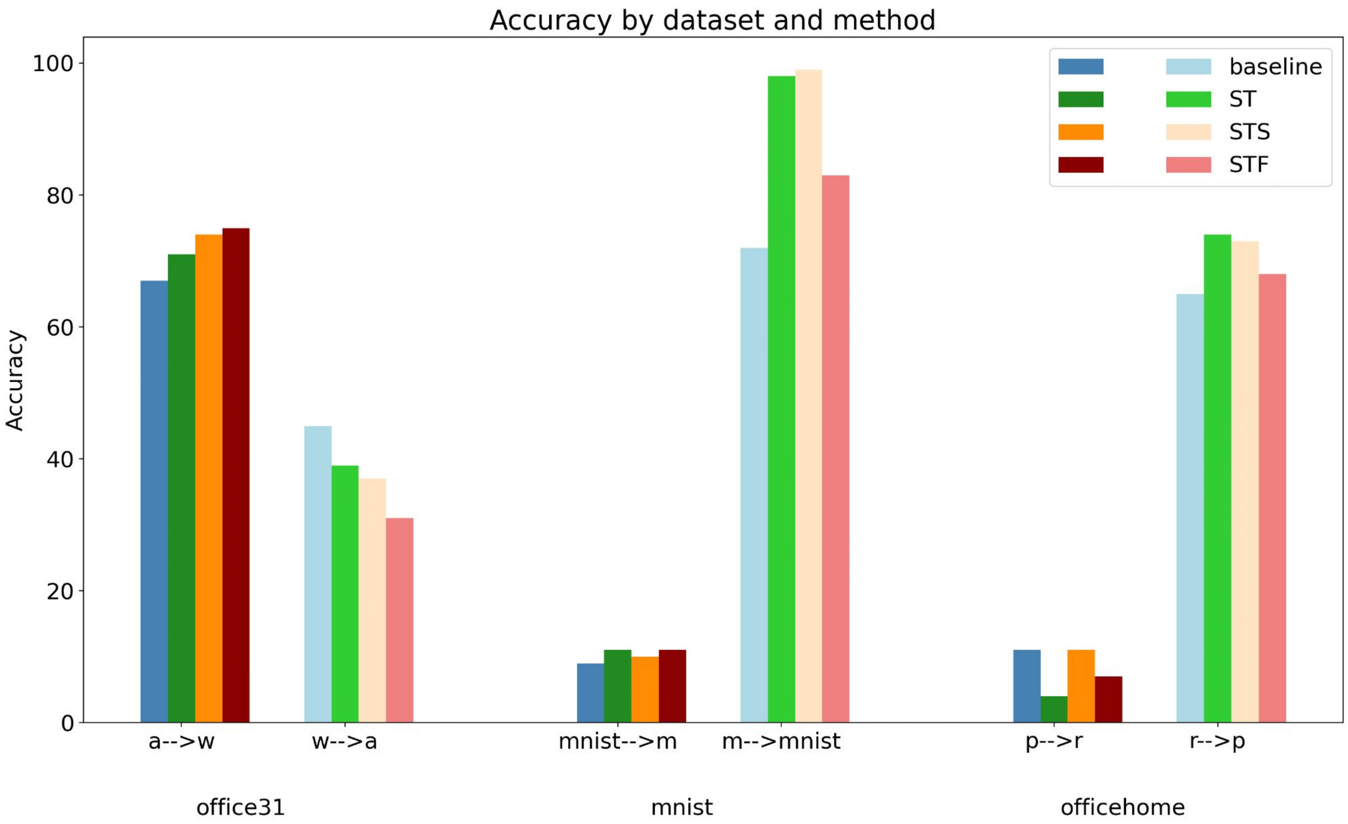
- AuxSelfTrain samples selection

    - *target* - highest **confidence** pseudo-label
    - *source* - **closest** to target distribution

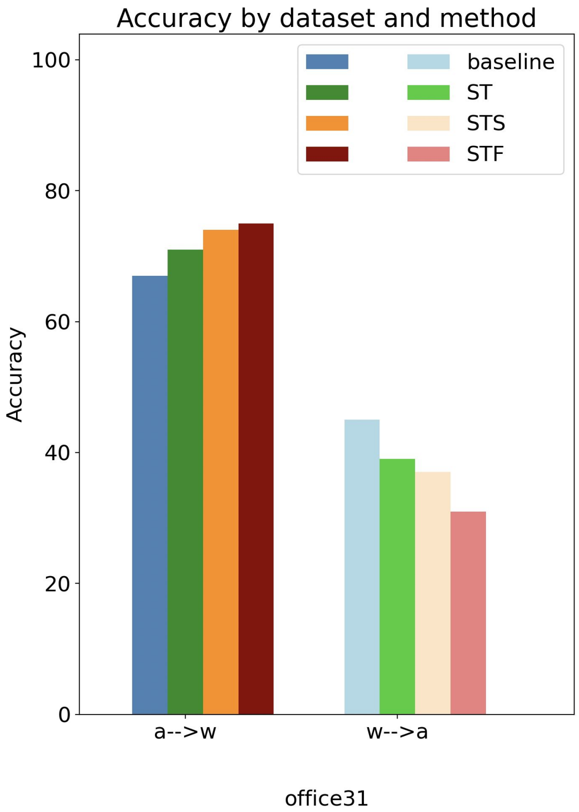
- Ablation studies:

    - *ST* - Full approach
    - *STS* - **source** samples are **randomly** selected
    - *STF* - both **source and target** are **randomly** selected

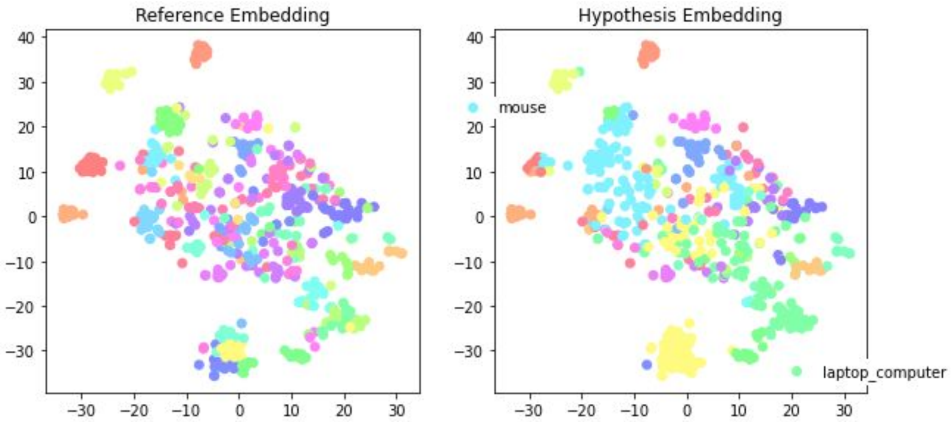
Results


Accuracy by dataset and method
Accuracy
100
80
60
40
20
0
baseline
ST
STS
STF
a-->w
w-->a
mnist-->m
m-->mnist
p-->r
r-->p
office31
mnist
officehome

Experiments: Office31

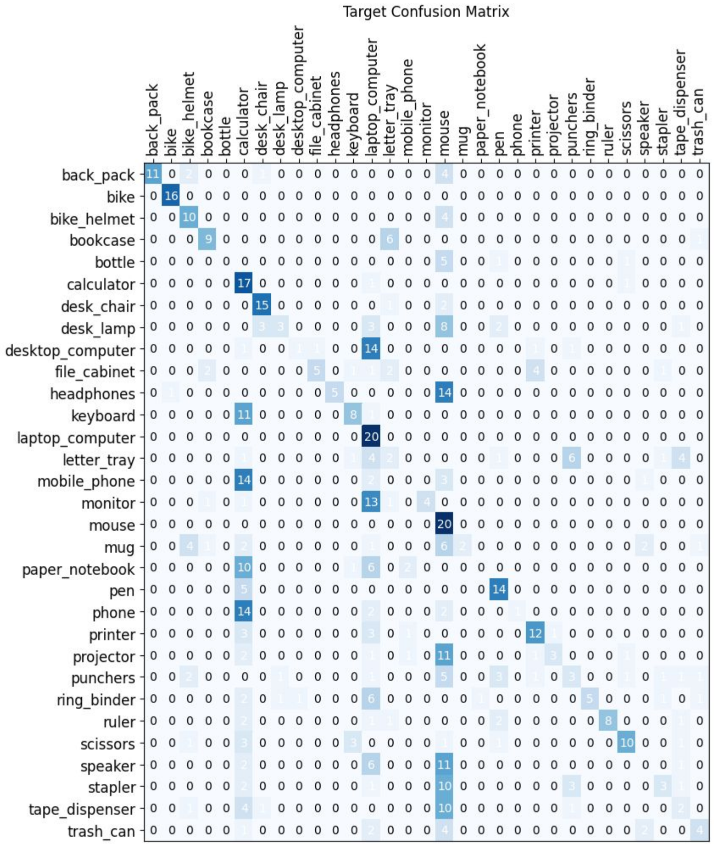


- Improvements in A -> W...
- ... But drop in W -> A
- Two problems:
  - clustering fails
  - some classes are over-represented

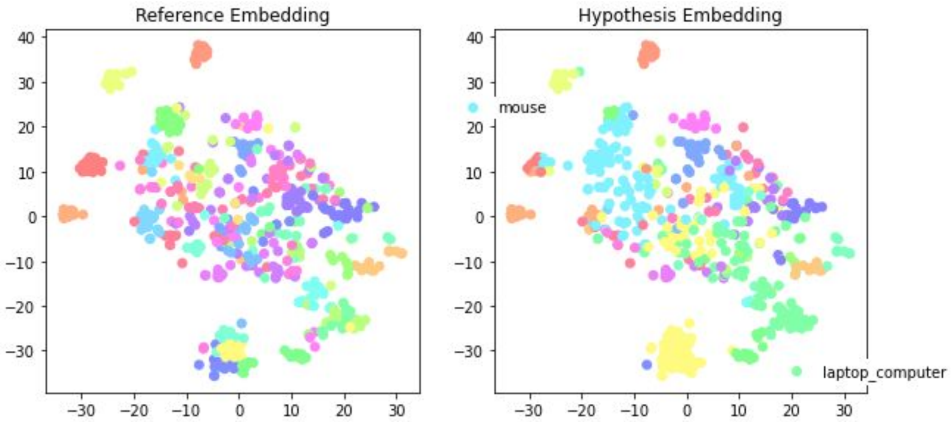


STS experiment

Experiments: Office31



- Hypothesis: unbalanced dataset
  - A: ~3000 samples
  - W: ~800 samples
- Perform experiments balanced dataset



STS experiment

Additional Datasets

MNIST

mnist-m | mnist

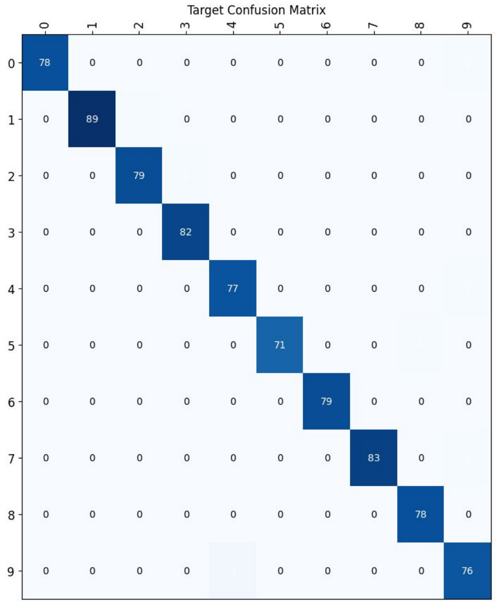
~4000 | ~4000

OfficeHome

Product | Real World

~2000 | ~2000

MNIST-M -> MNIST



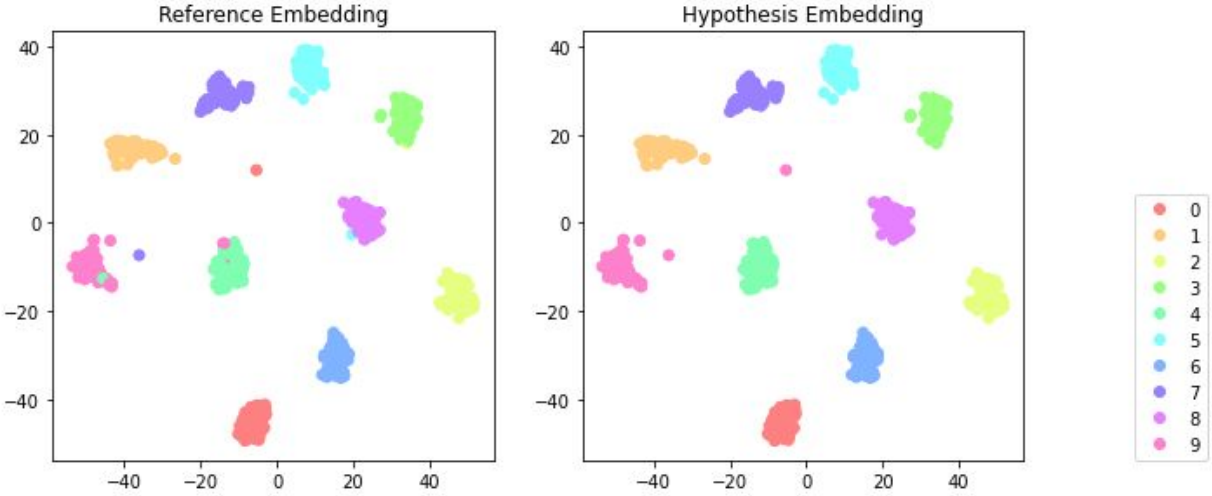
Target

Accuracy: 98.88

Loss: 0.00

Great!

Baseline is ~72%



MNIST -> MNIST-M

Target Confusion Matrix

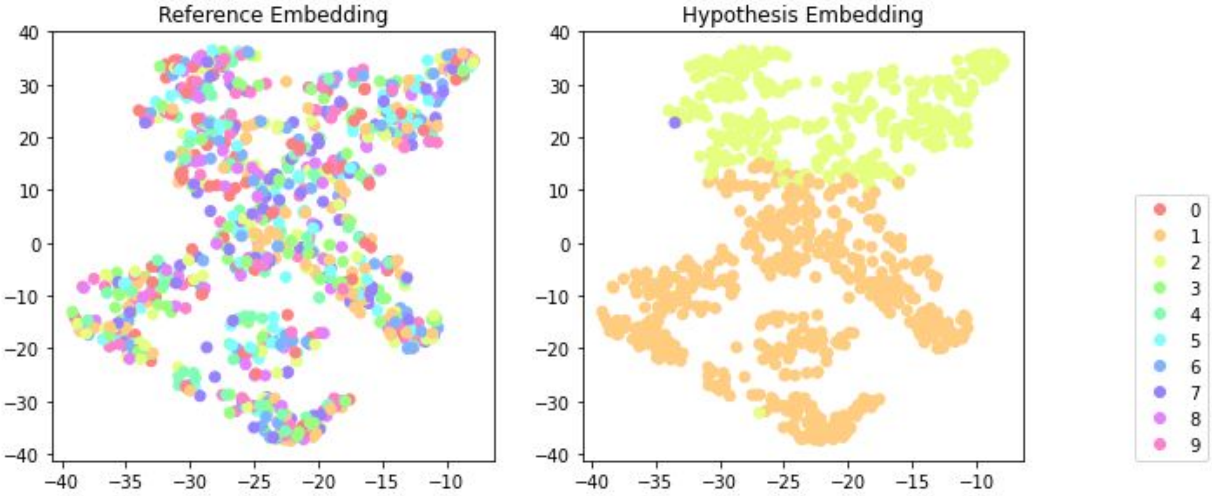
	0	1	2	3	4	5	6	7	8	9
0	0	45	44	0	0	0	0	0	0	0
1	0	71	31	0	0	0	0	0	0	0
2	0	59	32	0	0	0	0	0	0	0
3	0	62	31	0	0	0	0	0	0	0
4	0	54	35	0	0	0	0	0	0	0
5	0	45	37	0	0	0	0	0	0	0
6	0	49	40	0	0	0	0	0	0	0
7	0	59	34	0	0	0	0	1	0	0
8	0	54	35	0	0	0	0	0	0	0
9	0	53	37	0	0	0	0	0	0	0

Target

Accuracy: 11.45

Loss: 0.10

Yikes!



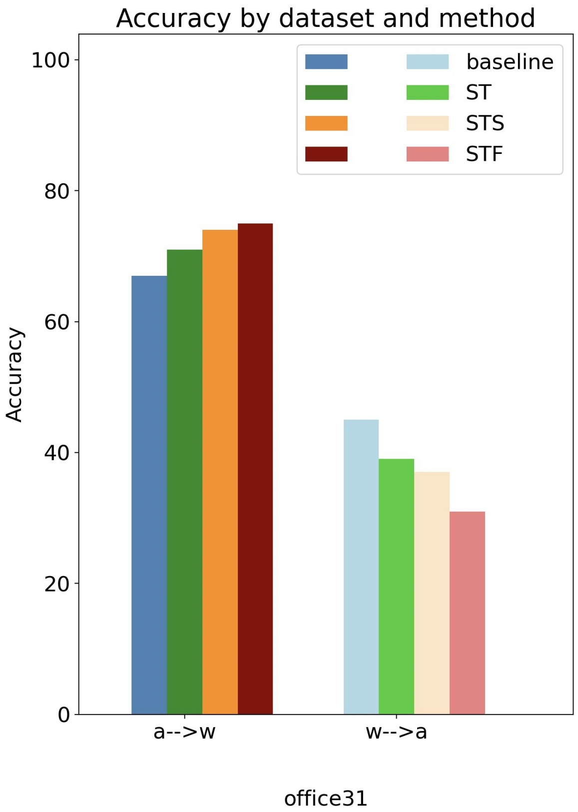
Asymmetric domains

- Results show similar patterns in OfficeHome
  - **R -> P** - good results
  - **P -> R** - no results

- Probably due to an **asymmetric** domain shift
  - MNIST-M is MNIST but with **more** information
  - same for R and P

- The model only work with **small** domain shift
  - otherwise, it is **over-confident** on one single label

Back to Office31



Why does it perform "suspiciously well" on A -> W?

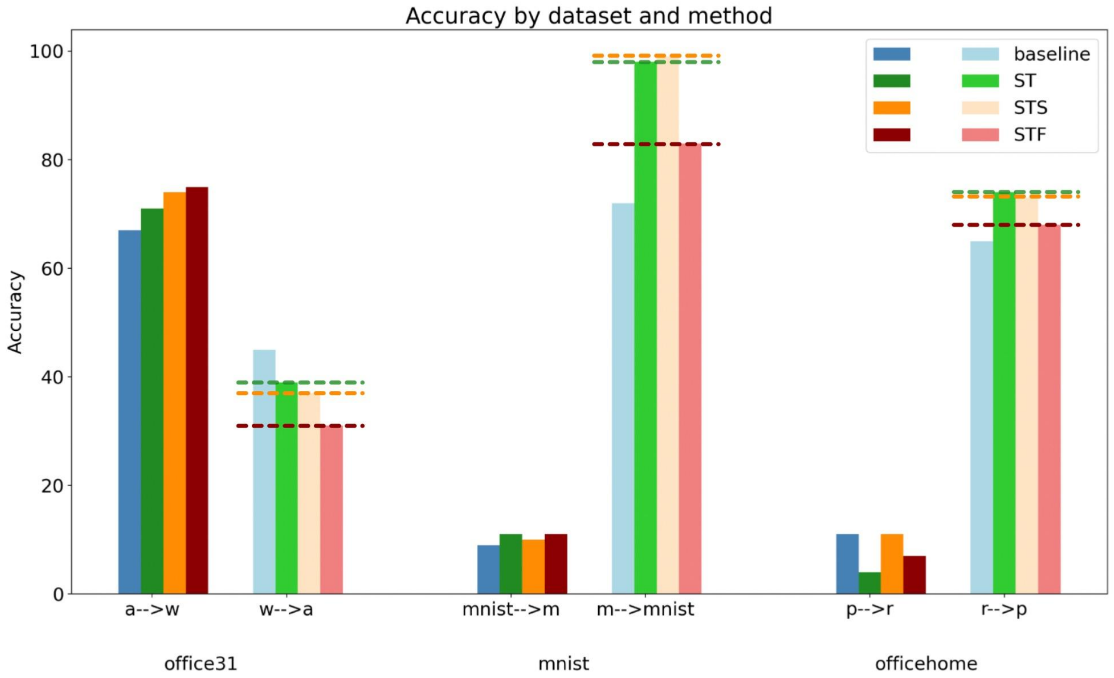
- amazon has some webcam-style images

Why does it perform "suspiciously bad" on W -> A?

- the domains are not balanced!

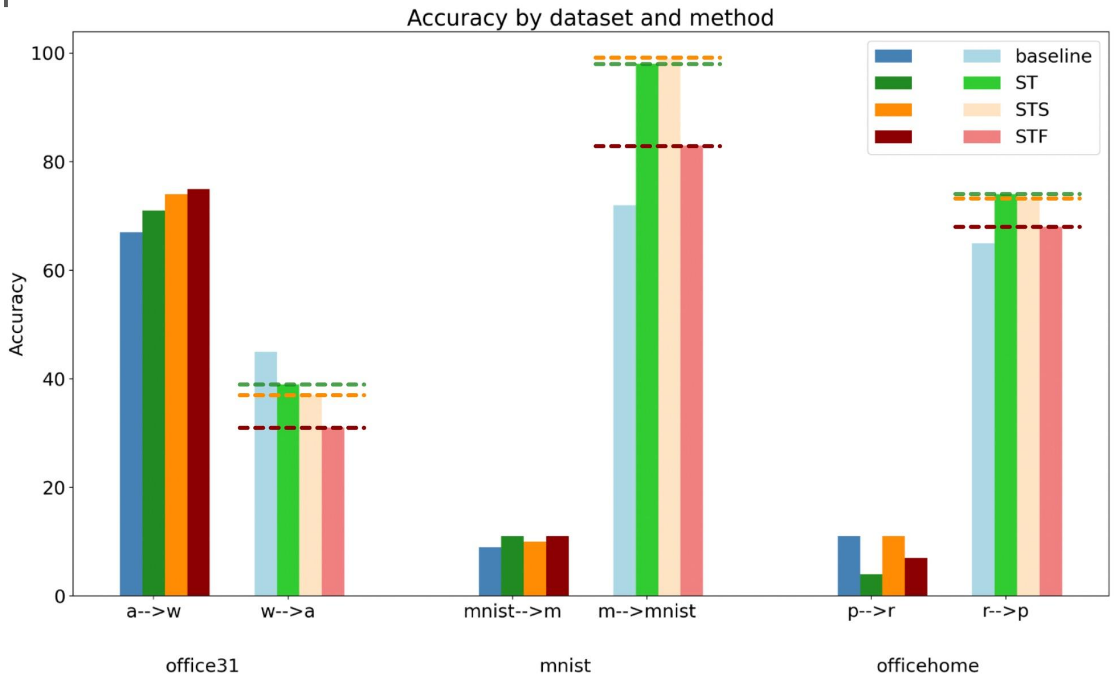
Ablation study

- ST, STS ≫ STF
    - target sample selection **works**



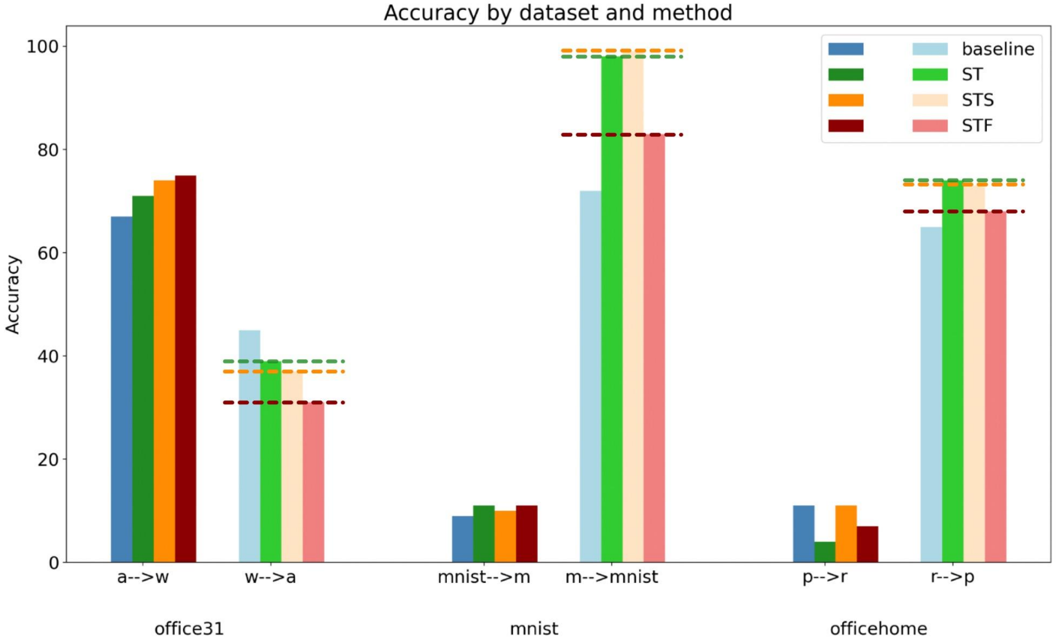
Ablation study

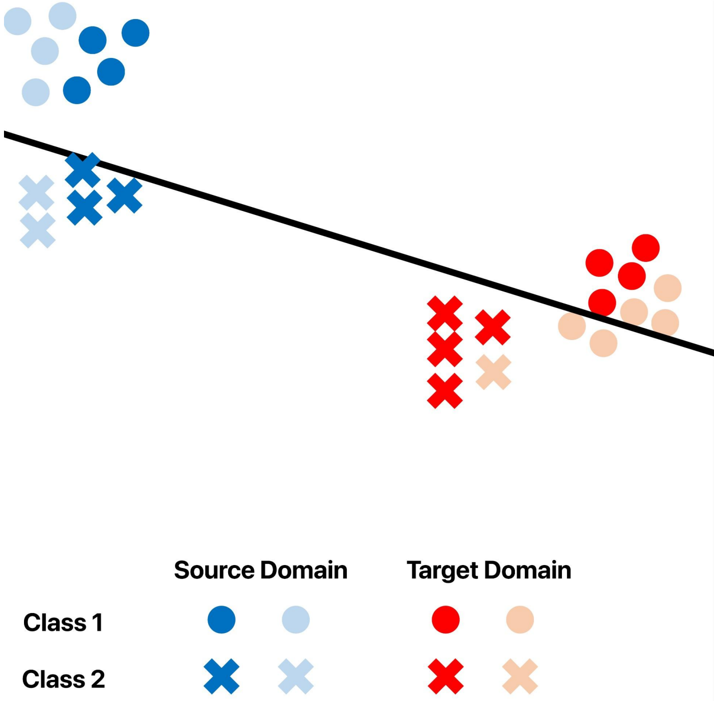
- ST ≅ STS
  - source sample selection does **not**
  - distributions are too far



Ablation study

- ST ≅ STS
  - source sample selection does **not**
  - distributions are too far





Future works

- Implement **model ensemble** on the target source samples selection strategy
  - requires significant computational power

- Behaviour on MNIST -> MNIST-M resembles the **mode collapse** problem of GANs
  - use toolchains from GAN literature to further explore
    - e.g. batch discrimination
  - add penalty/threshold when few classes are over-represented

- Test our DANN methods on other datasets
  - provide insights on your model

Conclusion

- **Avoid** to **over complicate** the model
- **Keep** the model **simple** but exploit it better (change losses and/or optimizer)
- Some models require careful **fine-tuning** of hyperparameters
- There is **no panacea** model
    - A thorough dataset exploration is crucial
    - Pick the best DA approach given the dataset

Thank you for your attention

Thank you for your attention

…and merry Christmas!