Robust Domain Adaptation

8 ECTS Semester Project - CSE

Eelis Mielonen

Professor: Volkan Cevher

Supervisor: Grigoris Chrysos

Laboratory for Information and Inference Systems

Ecole Polytechnique Federale de Lausanne

Motivation

- One problem that machine learning practitioners may encounter is a decrease in the performance of a model when the input data distribution changes.
- You can try to retrain continuously on the target domain, but you can't always obtain labels.
- Domain Adaptation and transfer learning methods have been developed to address this problem.
- We're interested in the case where we have no labels in the target domain, ie. **Unsupervised Domain Adaptation (UDA)**

Examples

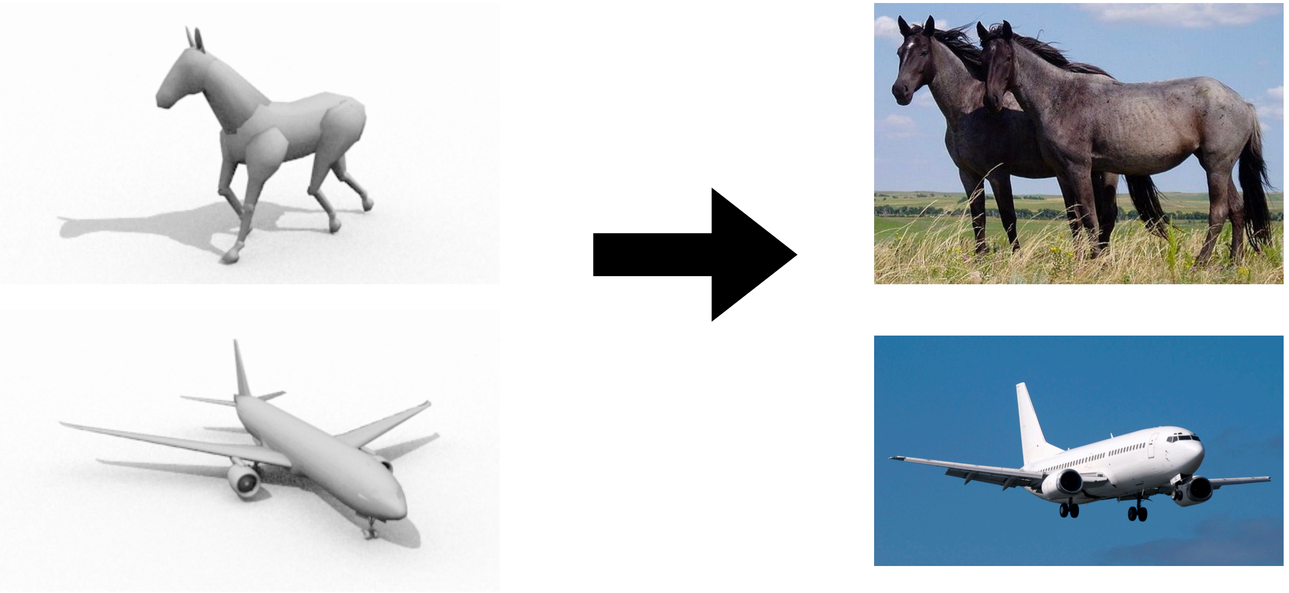
- MNIST to MNIST-M
- Model Trained on Classical MNIST performs worse on MNIST-M

Source

Target

Examples

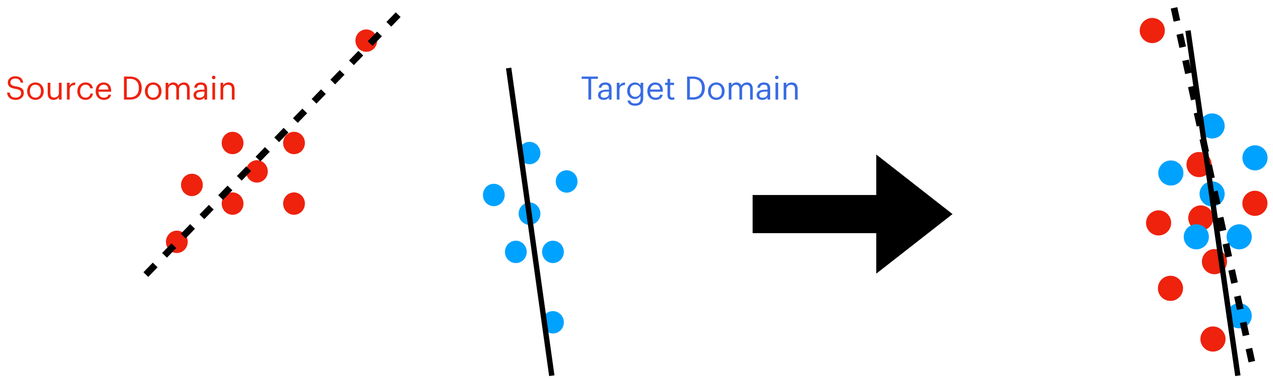
- Visda2017: 3D models to real images



Approaches in Unsupervised Domain Adaptation

- There are several techniques for image classification in unsupervised domain adaptation
- Best method may depend on the specific application and the characteristics of the data.
- Transfer Component Analysis, Adversarial Domain Adaptation, Self-learning...
- Some of the most successful “classical methods” in UDA for computer vision employ adversarial training.

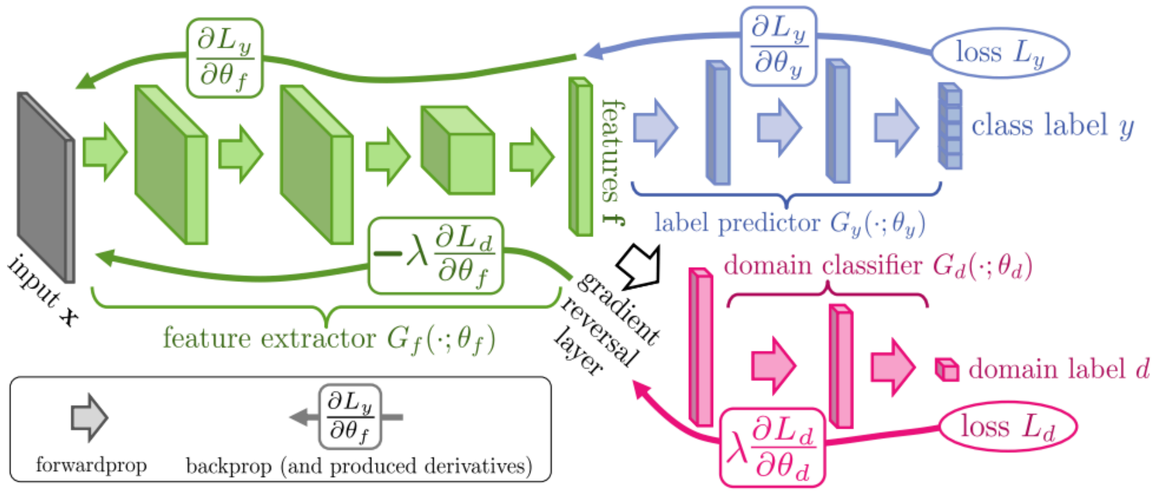
Key Idea : learn features that are invariant between domains, but which discriminate classes



UDA with adversarial learning

Adversarial methods balance two objectives in a min-max game:

1. **Maximising** domain confusion / minimising domain discrepancy
2. **Minimising** model classification loss



- Domain-Adversarial Training of Neural Networks (2015 - ICML)
- Add a classifier head for predicting domain label.
- **Minimize** Classification Loss in Source Domain.
- **Maximise** domain classification loss using unlabelled target and source images.

UDA with adversarial learning

- Generalisation bounds exist for target domain risk:

$$\epsilon(h)_{\mathscr{D}_T} \leq \hat{\epsilon}(h)_{\mathscr{D}_S} + \mathscr{A}(\mathscr{D}_{\mathcal{S}}, \mathscr{D}_{\mathcal{T}}) + \lambda$$

- Intuition: we can minimise source domain risk and a domain discrepancy term to bound the target domain risk.
- DANN and a other methods that followed borrowed this intuition. Minimise source classification loss and the distance between the two data distributions.

Issues with UDA: Worst Performing Classes

- Observation: the weakest classes in the source domain are likely to be the weakest classes in the target domain.
- As expected, relative differences in per-class accuracies are amplified in the target domain.

	MNIST (Source)	**MNIST-M (Target)**
Weakest Classes	Nine, Eight	Eight, Nine
Performance Gap	1.2%	2.5%

	Visda Synthetic (Source)	**Real (Target)**
Weakest Classes	Car, Bus, Truck	Bus, Car, Truck
Performance Gap	3%	43%

Performance Gap = Best Class Accuracy - Worst Class Accuracy

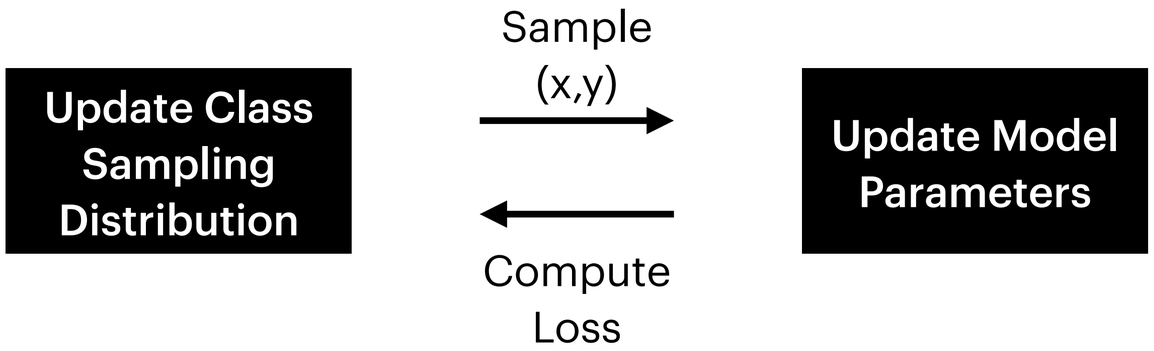
Methods

CFOL

- In adversarial training, even when the average accuracy of the model is acceptable, some classes can be noticeably worse than others. **Not ok** in some applications.
- **Class Focused Online Learning:** Instead of minimising the average risk, we minimise maximum **class conditioned** risk (ie. the weakest link):

$$\min_{\theta} \mathbb{E}_{(x,y)\sim P(x,y)}[L(\theta, x, y)] \quad \Rightarrow \quad \min_{\theta} \max_{y\in[k]} \mathbb{E}_{x\sim P(\cdot|y)}[L(\theta, x, y)]$$

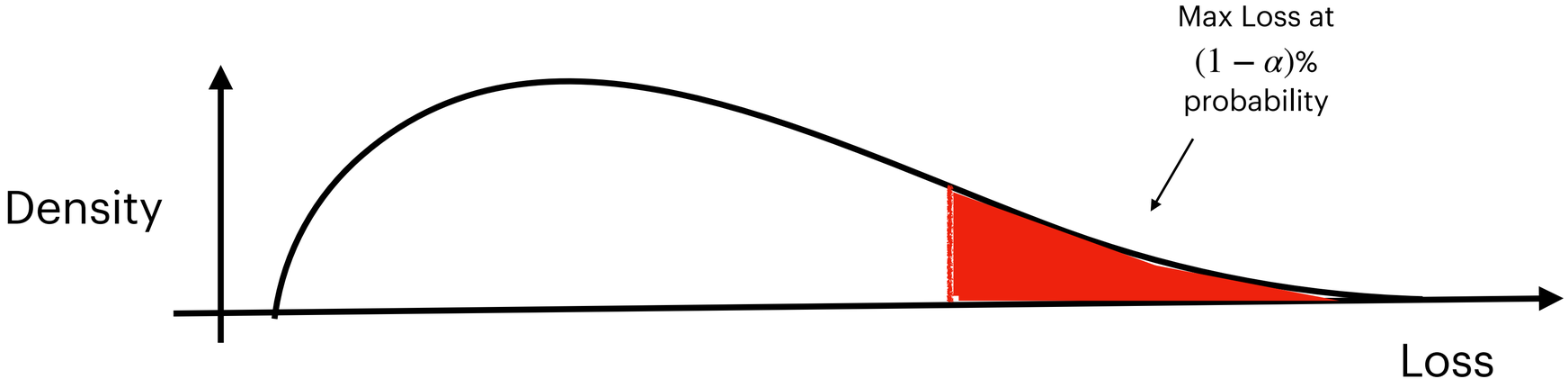
- This objective can be minimised by cycling between two steps:



Methods

LCVAR

- We can also consider a different robust loss : Conditional Value at Risk (CVAR)
- Explained through example: if the CVaR for a portfolio is $1 million at a 99% confidence level, it means that there is a 1% chance that the expected loss equals $1 million.
- CVAR at a confidence level $\alpha \in [0,1]$ measures the expected loss at the tail end of the loss distribution.



Methods

LCVAR

- How can we optimise this w.r.t classes?
- For discrete random variable we have the expression:

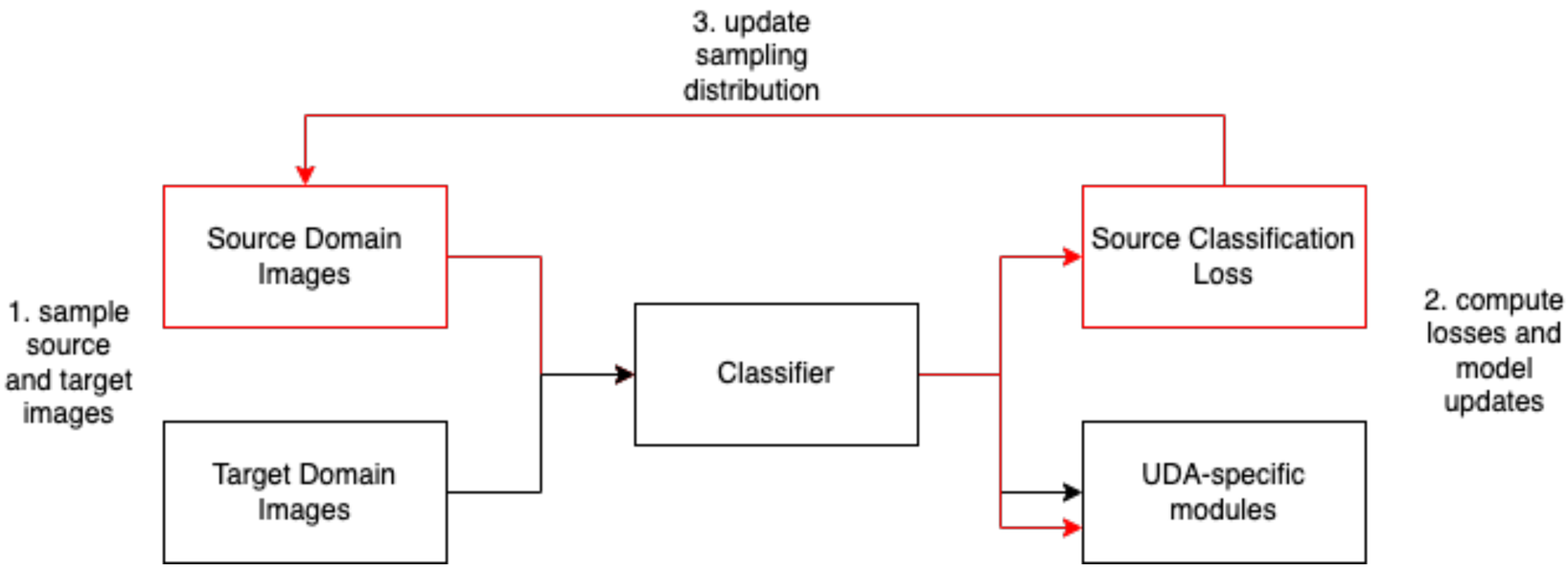
$$\text{CVAR} = \sup_{q \in Q_\alpha} \sum_{i=1}^{k} q_i p_i c_i$$

- Replace c_i with loss of class i, to obtain Label CVAR (Label Conditional Value at Risk)
- Resulting dual optimisation problem can be optimised with back-propagation:

$$\inf_{\lambda \in \mathbb{R}} \{ \frac{1}{\alpha} \sum_{i=1}^{K} p_i \text{ReLU}(L_i(\theta) - \lambda) + \lambda \}$$

Combining with UDA approaches

- The obvious approach is to use a reweighed source domain classification loss (with CFOL, LCVAR).
- Question remains: is focusing on the worst classes in the source domain beneficial for generalisation performance (target domain accuracy)?

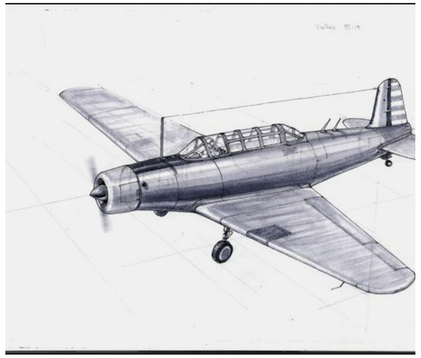


Selected UDA approaches

- Minimum class confusion (MCC): add a “class confusion” regularisation term.
- Margin Disparity Discrepancy (MDD): Minimise a computable version of domain discrepancy “induced” by the classifier, involving a min-max game over the model parameters.
- Conditional Domain Adversarial Networks (CDAN): Makes use of domain adversarial training like in DANN.

Experiments

- Fine-tune a Resnet101 on the source domain images, simultaneously perform domain adaptation.
- **Benchmarks: Visda2017 & DomainNet.**
- **DomainNet** is a large scale dataset with 5 different domains and over 300 classes. Ideal for computing class accuracy distributions.
- Simple Experimental set up: turn on CFOL and LCVAR with different tuning parameters (uniform vs. Adaptive distribution), and measure the effect.



→

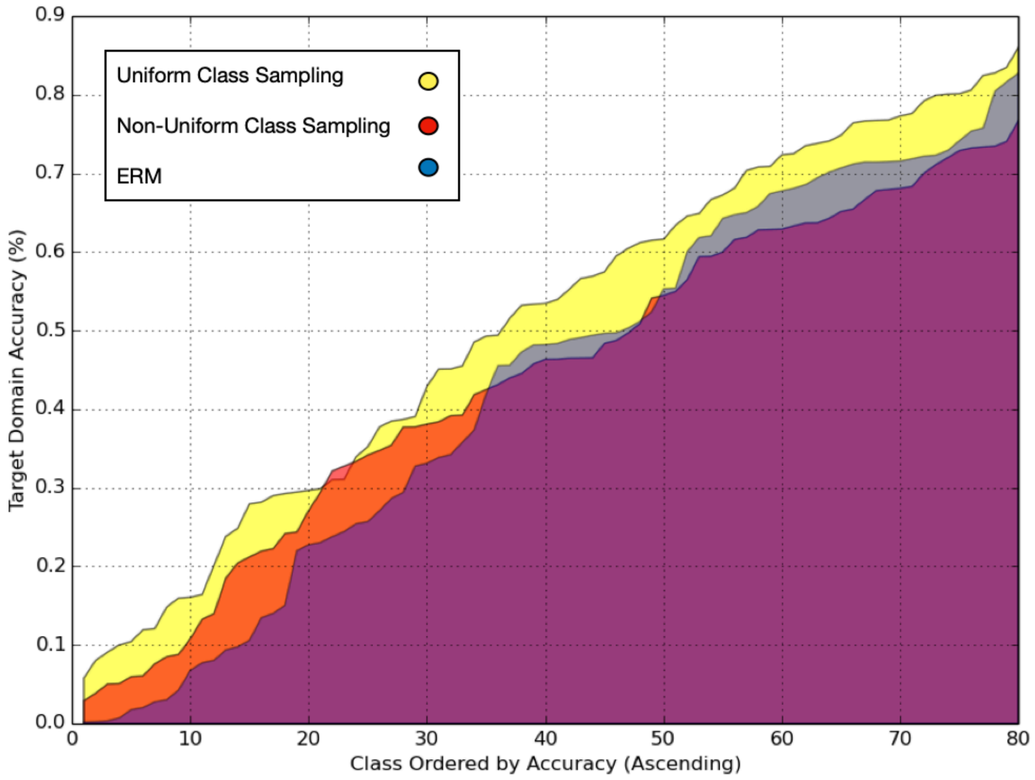
Experiments

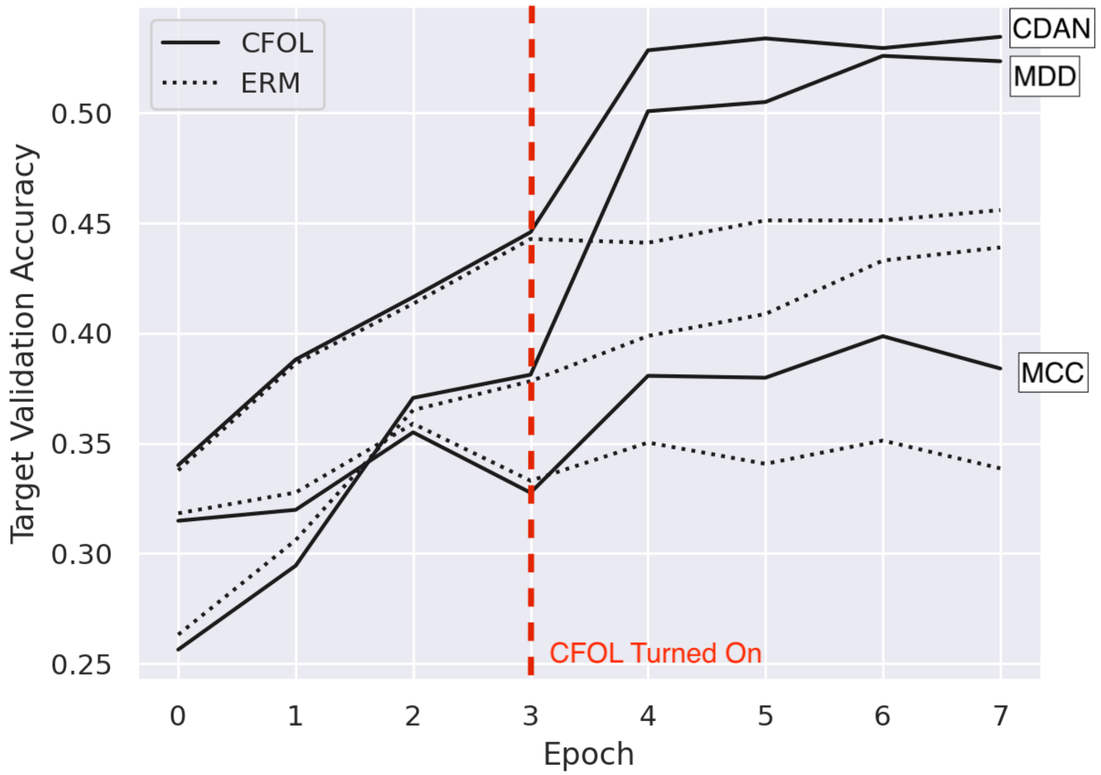
Results - DomainNet

- If we set $\eta = 0.001$ and change the sampling distribution aggressively, we don't gain much with CFOL.
- Also, setting $\alpha = 0.5$ with LCVAR seems to have no effect.
- Most often sampling the classes uniformly lead to favourable results with CFOL. In some cases it sped up training / improved class accuracies, without impacting average accuracy in the source domain.
- With LCVAR, setting $\alpha = 0.8$ (focus on 20% worst classes) improved worst class performance, but at the expense of the best classes.

Experiments

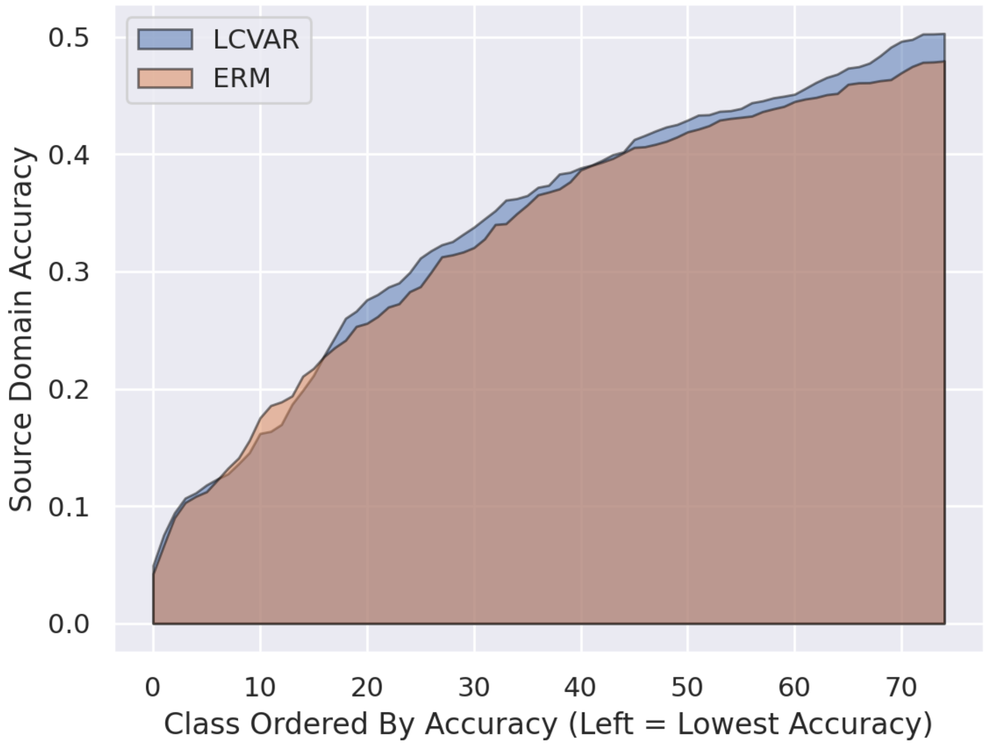
Results - CFOL on DomainNet

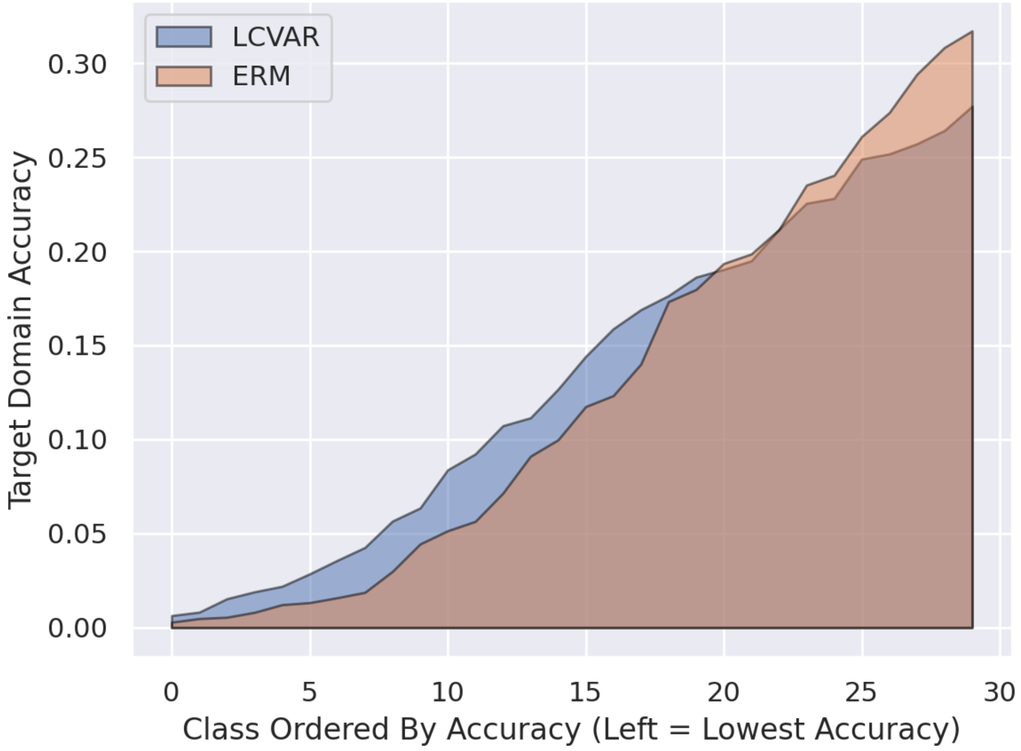




Experiments

Results - LCVAR on DomainNet

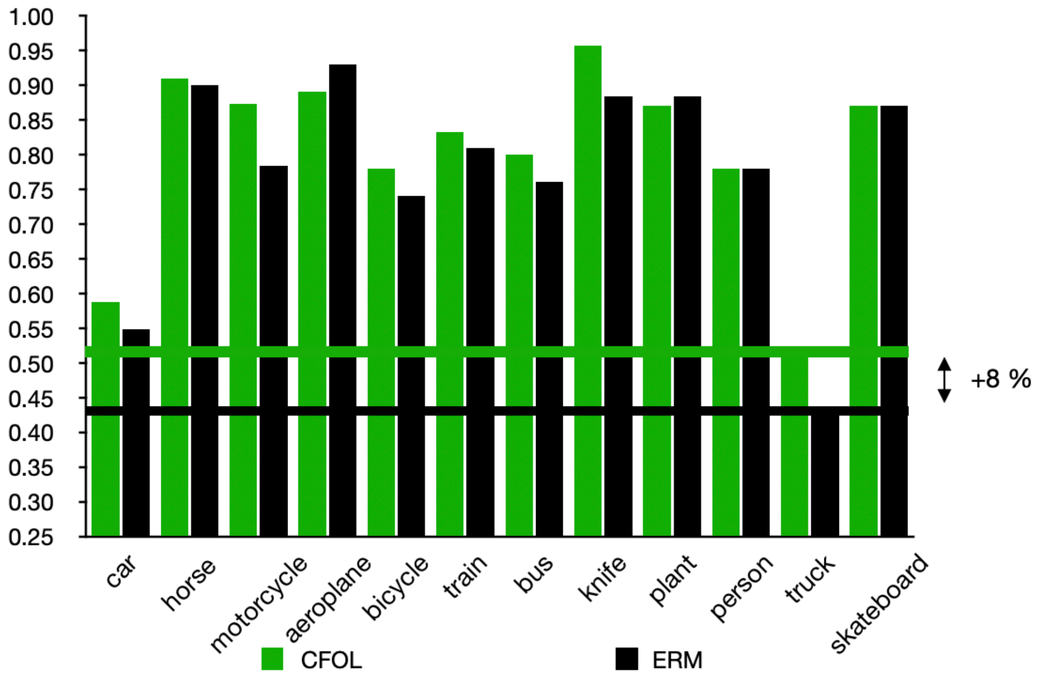




Experiments

Results - VisDA2017

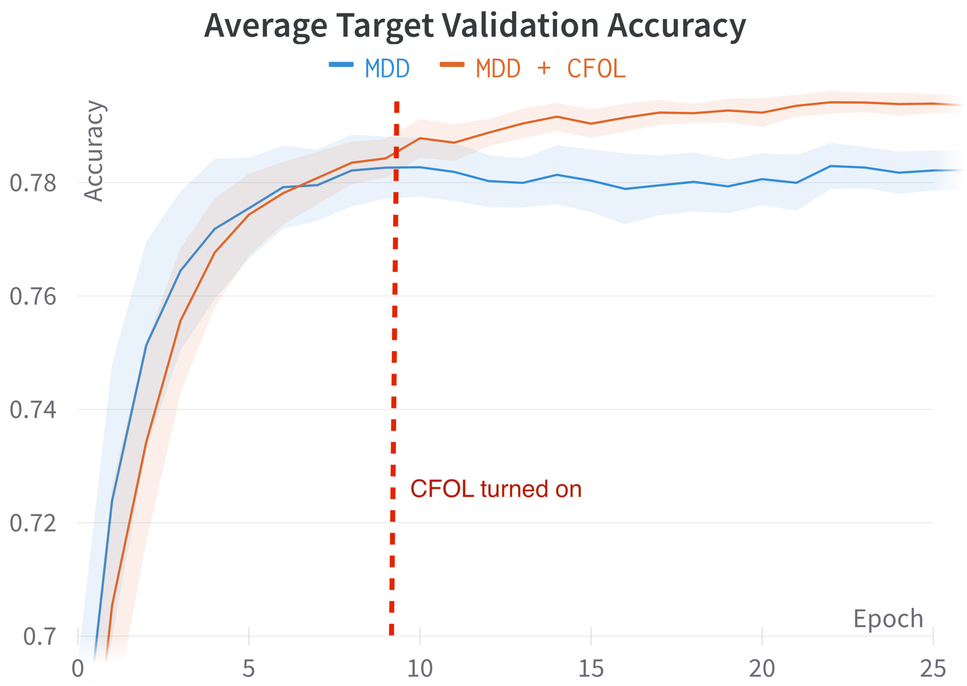
- For this dataset nothing happened with LCVAR
- However, we measured an average improvement in target domain accuracy using uniform class sampling with CFOL.

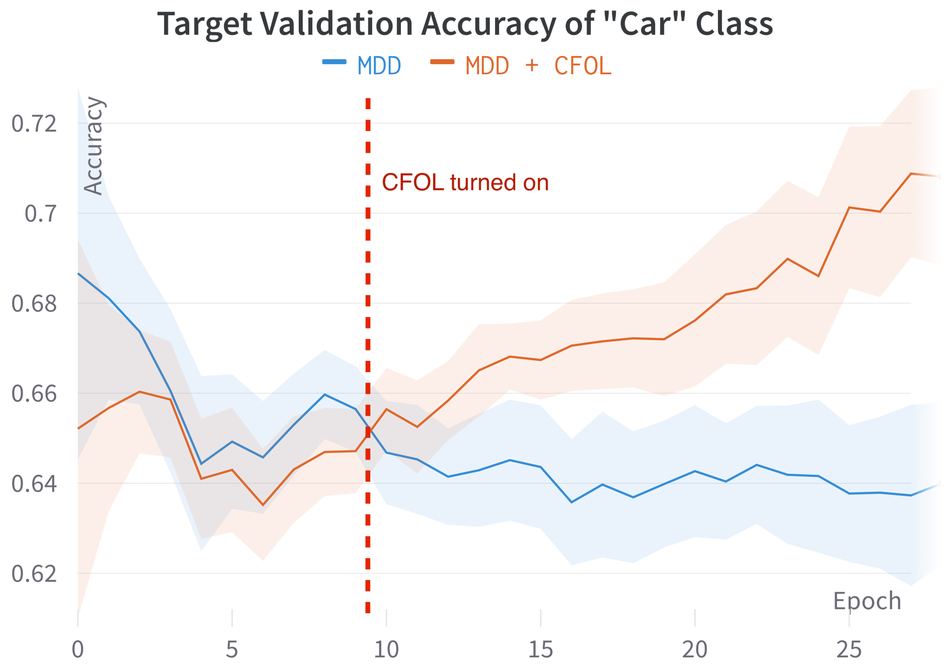


Minimum class accuracy recorded across methods improved by 8%

Experiments

Results - VisDA2017





Uniform Sampling

- Why do we see an improvement with uniform sampling, and why does the improvement come from sampling in the same class?
- The model only needs to compare batches of images where the domain is different, not the class. By sampling this way we inject the inductive bias that the distribution discrepancy doesn’t arise from class differences.

Conclusion

- It’s possible to alter the generalisation performance of models by re-weighting the source domain risk.
- Surprisingly, for Visda it is more beneficial to sample batches from the same class.
- Need to gather more statistics over multiple domain transitions.
- Could repeat the study in a controlled toy-dataset setting to build a better understanding.
- Also lacking: ablation study.

Thank You!