https://arxiv.org/pdf/2106.07780.pdf

KL GUIDED DOMAIN ADAPTATION

A. Tuan Nguyen* † Toan Tran ‡ Yarin Gal †

Philip H. S. Torr † Atılım G¨unes¸ Baydin †

Empirical Risk - When we compute the **loss** between the **target** and our **prediction**

- **Empirical Risk Minimisation** - Minimising **loss function** to minimise error between targets and predictions
    - A common training procedure

Domain Adaptation / Transfer Learning - Accounting for training data and testing data having differing distributions

- ERM doesn't work well for this kind of task as it assumes the two distributions are the same
- It is the study of solving the **domain shift** between the source and target domains, to reduce data required

The paper derives a **generalisation bound** for the **target loss**, using the **training loss** and **reverse D_{KL}**

- They provide a bound for the maximum loss we are to see on the training set, based on the training loss, and calculating the reverse D_{KL} divergence, which measures how much information from our training would be useful for the testing performance
- This provides a loss that minimises the reverse D_{KL}, and hence better generalises to target domain

Distribution Alignment - learning a representation with a distribution that does not change between domains

- Two approaches: marginal alignment and conditional alignment
- **Marginal Alignment -** we can train with some training data that has the same **marginal** distribution over the source and target domain
    - It's about finding which occurrences in your source dataset corresponds the closest to the target dataset, then only using that data to train
- **Conditional Alignment -** align the conditional distribution of the label given the representation
    - Make it so that the distributions of our class labels matches between the source and target data

Extends from previous literature which only defined the reverse KL divergence bound for binary classification.

- KL divergence is estimated with samples, requiring no distribution alignment method (such as via minimax)
- Instead of training a feature extractor, you train a **probabilistic** feature extractor that gives you a distribution over which you can sample possible representations
    - *This is because the **output of the feature extractor are the variables of a Gaussian distribution** for each dimension in our features*
- During **test time**, your **classifier** then chooses a label based on the **average class** across **several sample representations**, which is fine, since we **only need to run inference once** to get a distribution, then sample it numerous times
    - The **classifier is small**, which means we can run it in parallel on our samples
-