Optimal transport application in Machine learning

Malte Algren

Previous lectures

Tried to connect it with previous lectures:

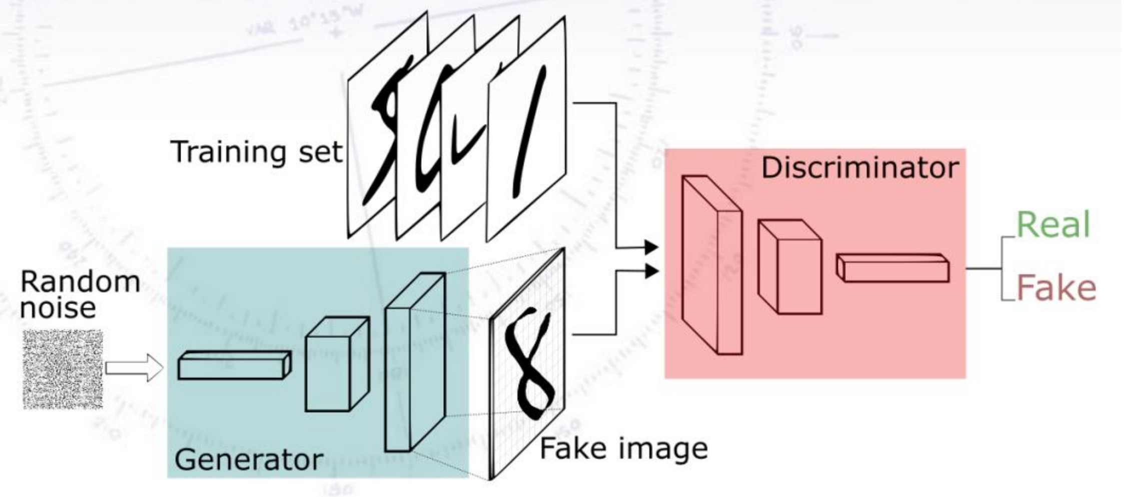
Week 4: [GANs]

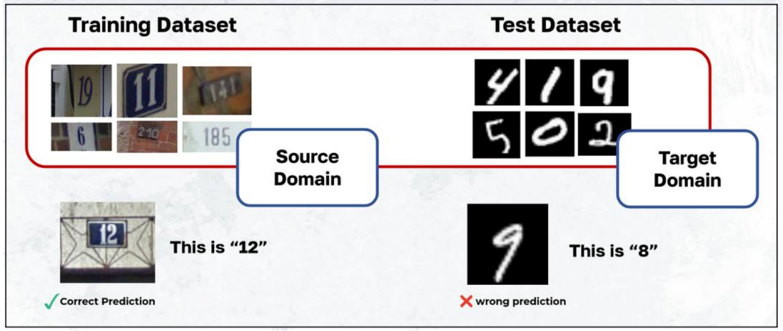
Week 6: [Domain shift]

Find a transformation T: $T_c(p(x|c)) = p(x)$

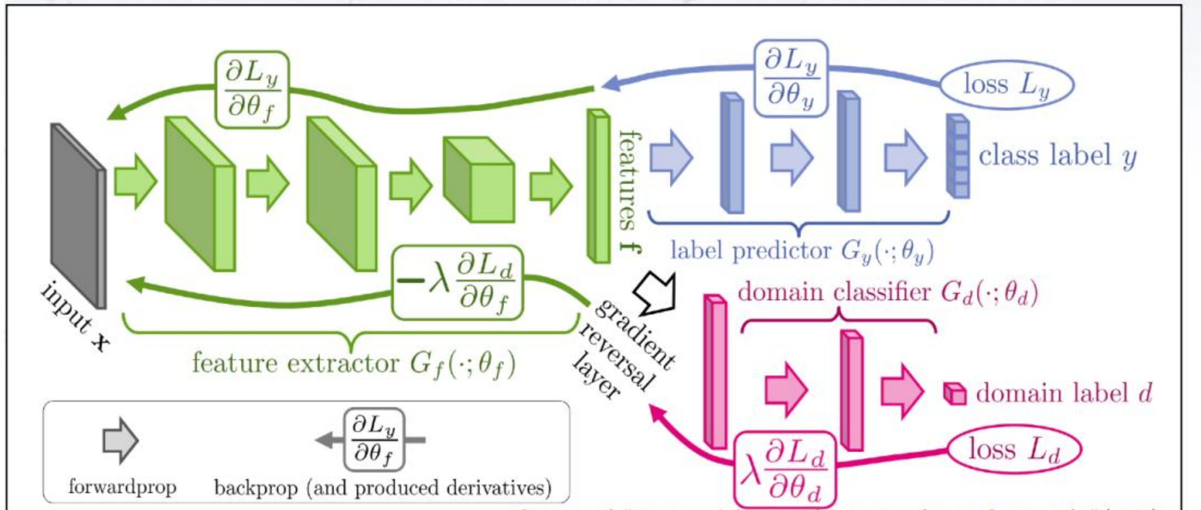
The discriminator / adversarial can also be seen as an addition to loss function, penalising (with λ) an ability to see differences between real and fake:

$$\text{Loss} = \text{Loss} + \lambda \cdot L_{\text{Adversarial}}$$





This problem is called "Domain Shift", i.e. there is **a "shift" between** the data that a model was **trained on** and the data that it was **applied to**.



Ganin et al. "Domain-Adversarial Training of Neural Networks" (2015).

The adversarial "forces" to learn from features that are common in domains.

11

Correct Domain Shifts (Domain Gap,

Domain Adaptation, Decorrelation or

Representation learning)

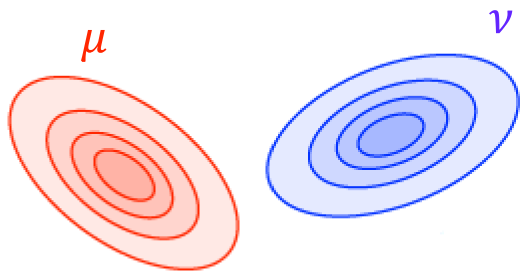
using Optimal Transport

Performance metrics

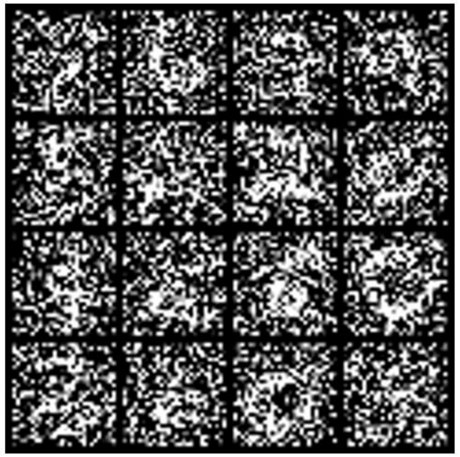
- How to measure convergence between μ and ν ?
  - BCE can be used to classify the difference

High dimensional image space

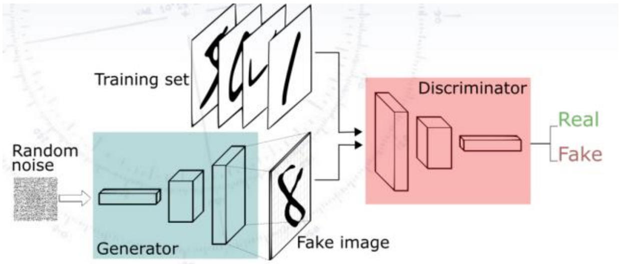
(Cat vs dog or generated vs true)



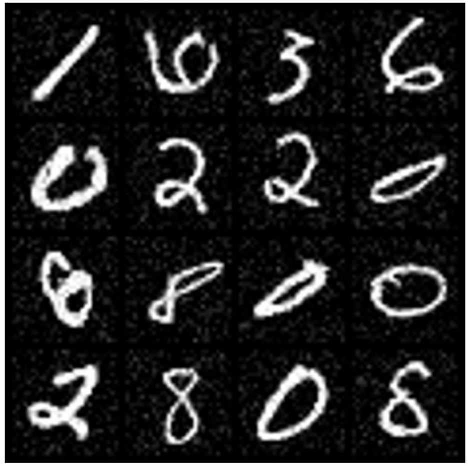
Random noise



GAN



Generated image (μ)

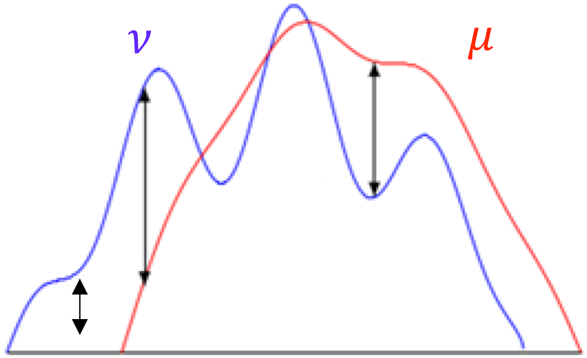


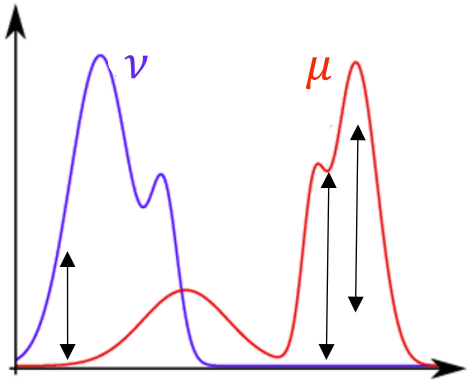
Target (ν)

Performance metrics

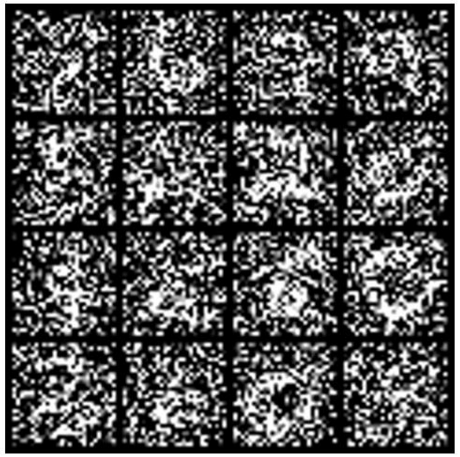
- How to measure convergence between μ and ν ?
  - BCE can be used to classify the difference
  - Project high dimension data into 1d simplex
  - Discriminator measure the ratio (vertical)

Can you think for a different measure?

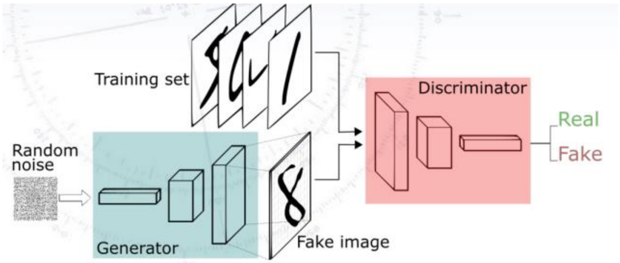




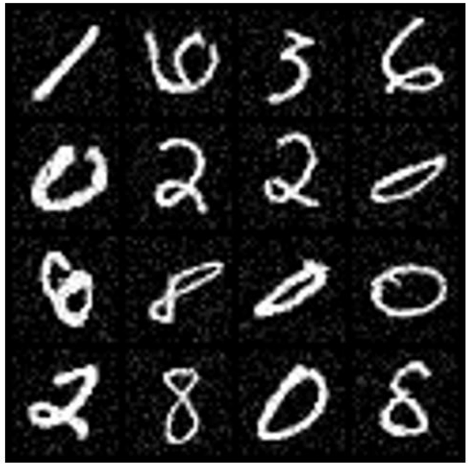
Random noise



GAN



Generated image (μ)



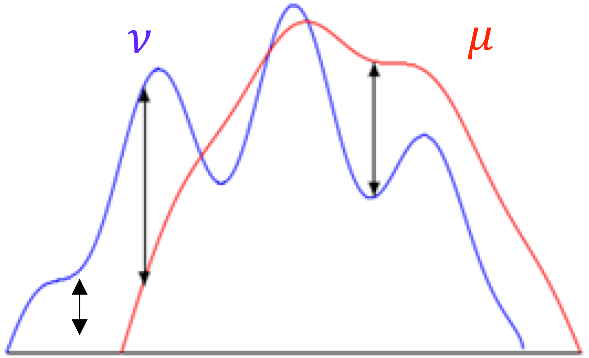
Target (ν)

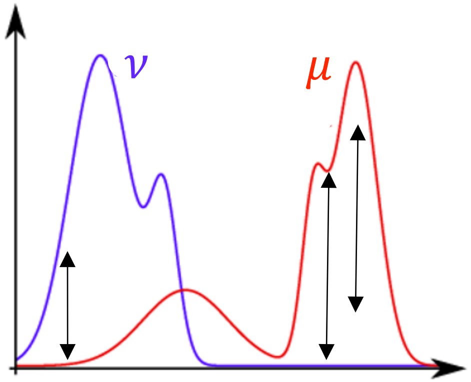
Distance metric – performance

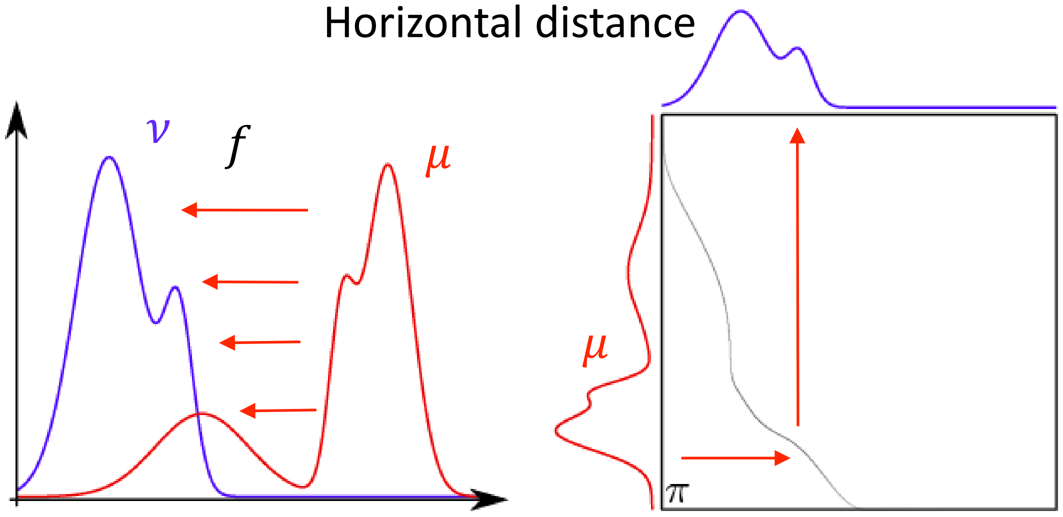
- How to measure convergence between μ and ν ?
  - BCE can be used to classify the difference
  - Project high dimension data into 1d simplex
  - Discriminator measure the ratio (vertical)

Can you think for a different measure?

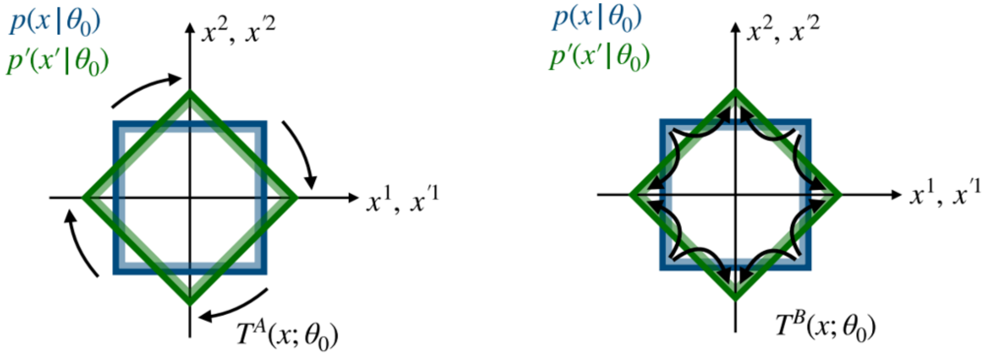
- Horizontal distance or transport distance
- Ratio > displacement vector
- **“Optimal transport”**







(a) Source and target (b) Transport map

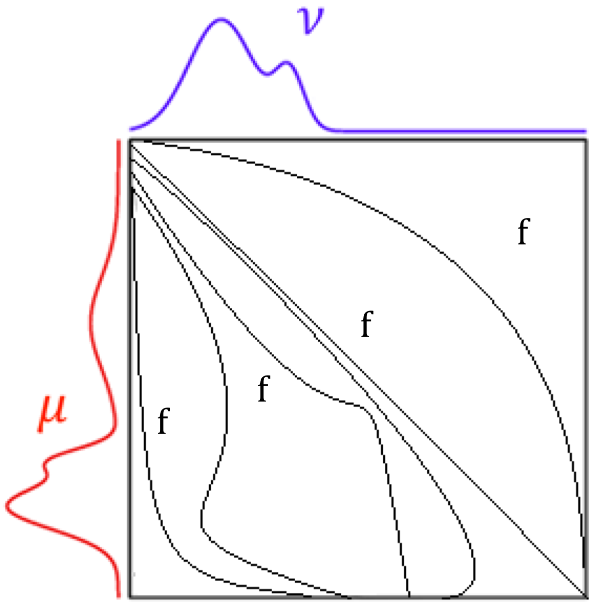


Transportation Theory

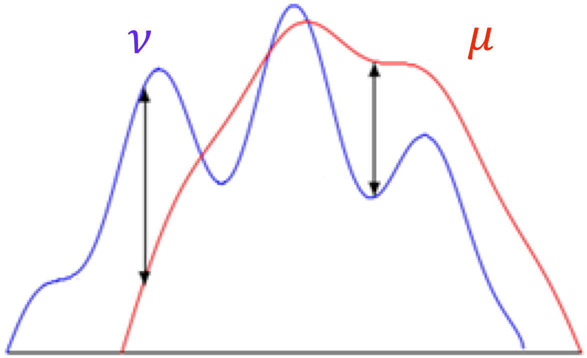
Wasserstein distance

- *Find the transport that minimizes the W*
  - $W_p(\mu,\nu) = \left(\inf_{f(\mu)=\nu} c(x, f(x))^p\right)^{\frac{1}{p}}$
- Combinatorial problem

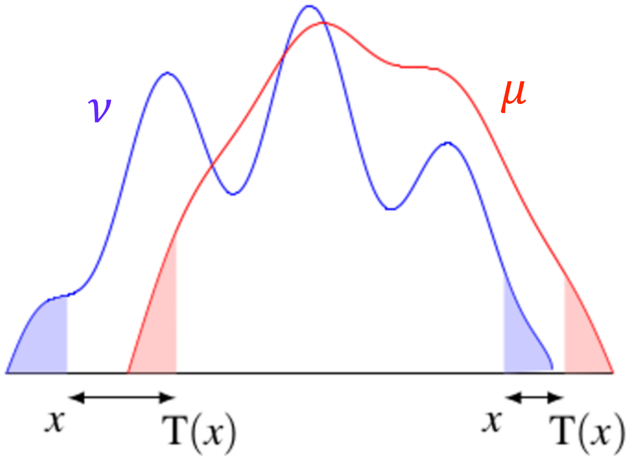
  "try" all transport maps
- Cost: $c(x,y) = \left(x - f(x)\right)^2$
  - By definition "order preserving"
  - Cyclical monotonicity



KL div

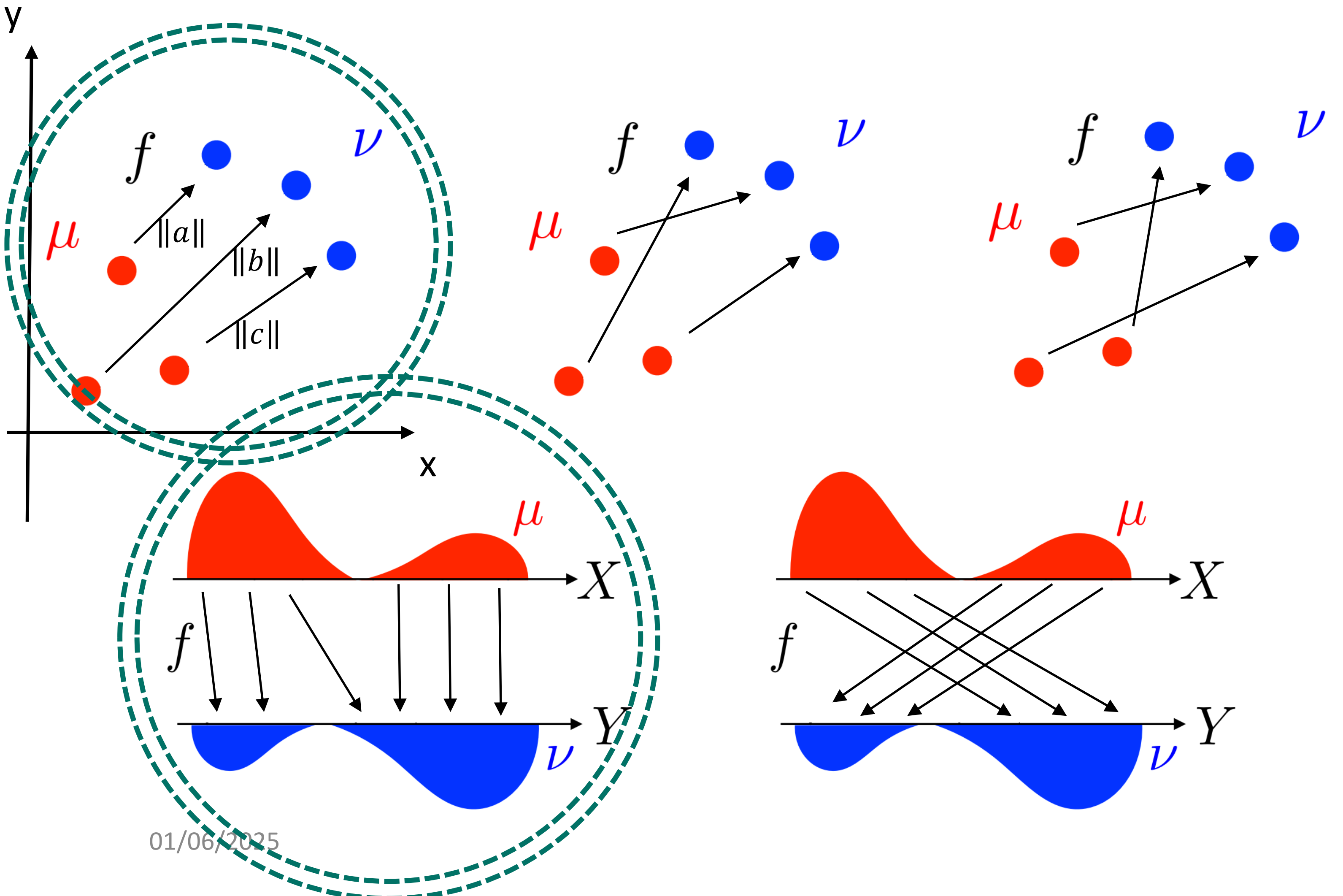


Wasserstein



The *optimal* in optimal transport

- Infinite number of possible transformation *f*
  - Which one to choose?



Optimal transport

$$\min_{\nu=f_\sharp\mu} \int_X c(x, f(x))\mathrm{d}\mu(x)$$

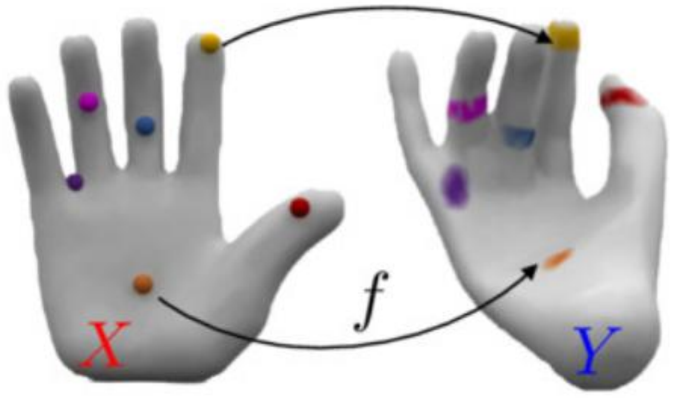
Select f with lowest c

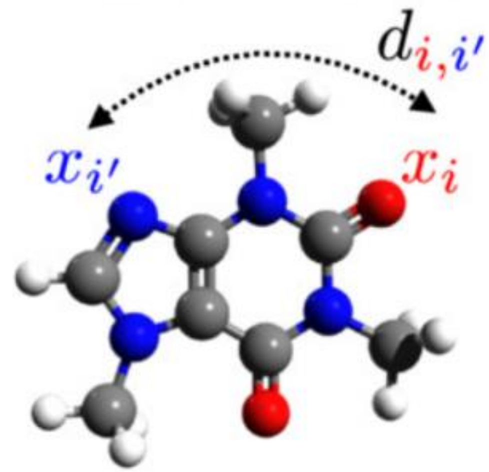
$$c(x,y) = \left(x - f(x)\right)^2$$

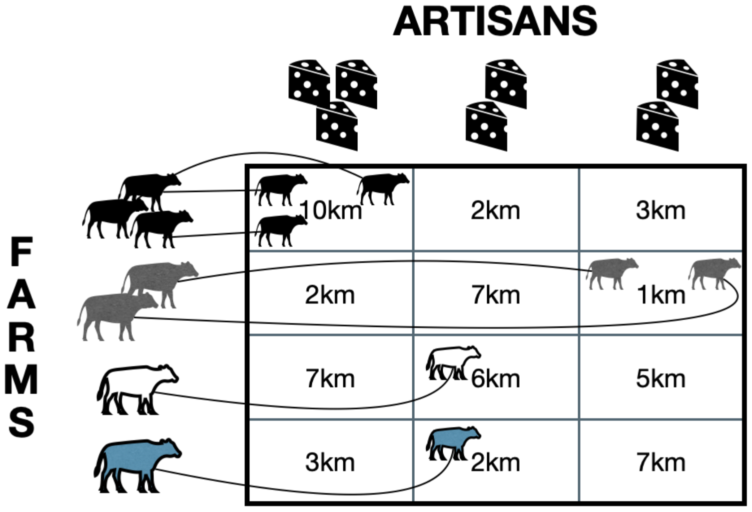
Optimal transport

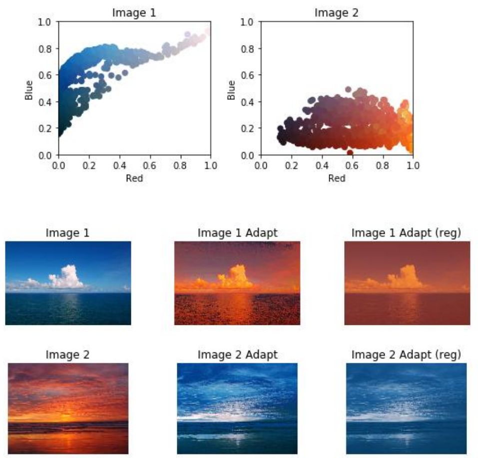
- Applications
  - Transport is costly
  - Minimum change is desirable

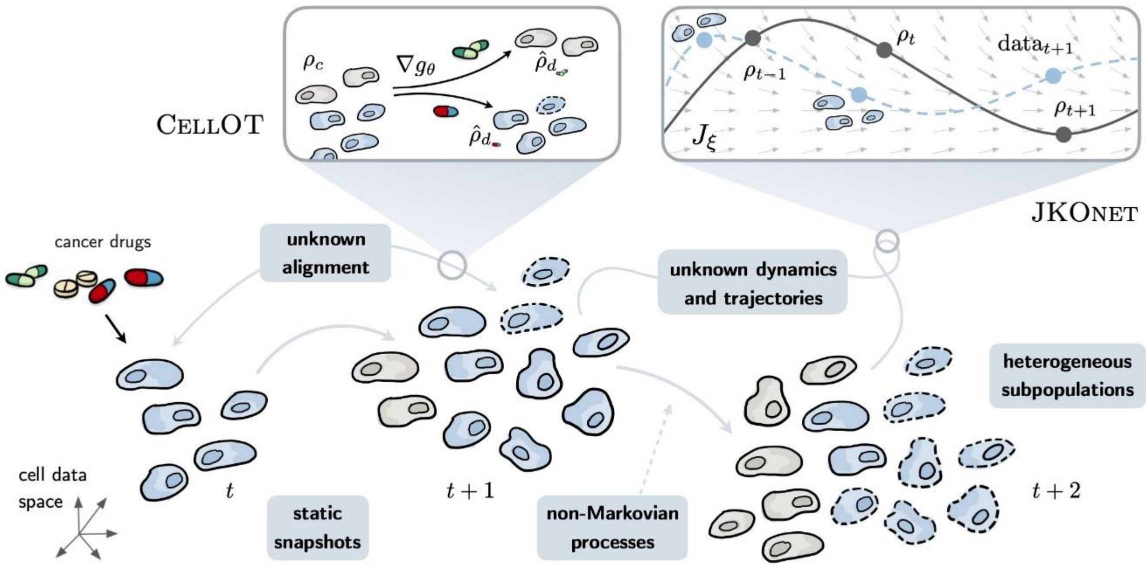
<u>Nature is lazy</u>











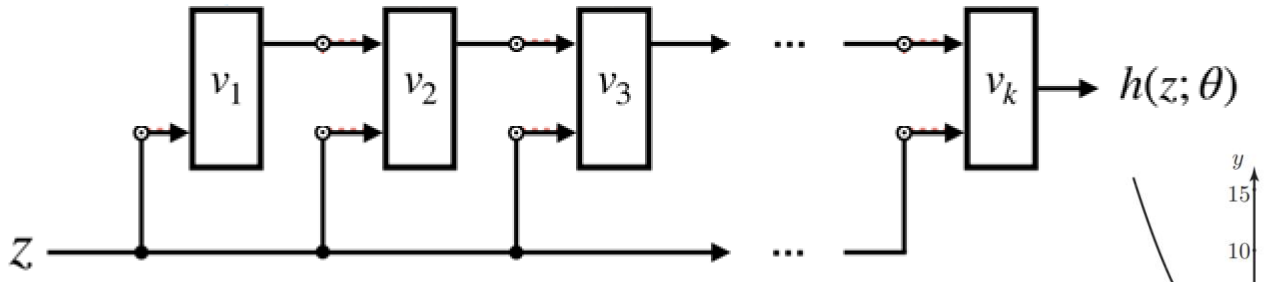
Find the optimal transport using machine learning

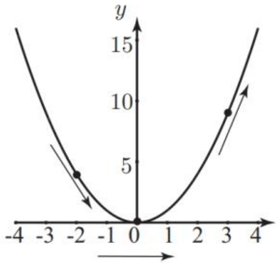
Solving Kantorovich dual problem

Optimal Neural Solver

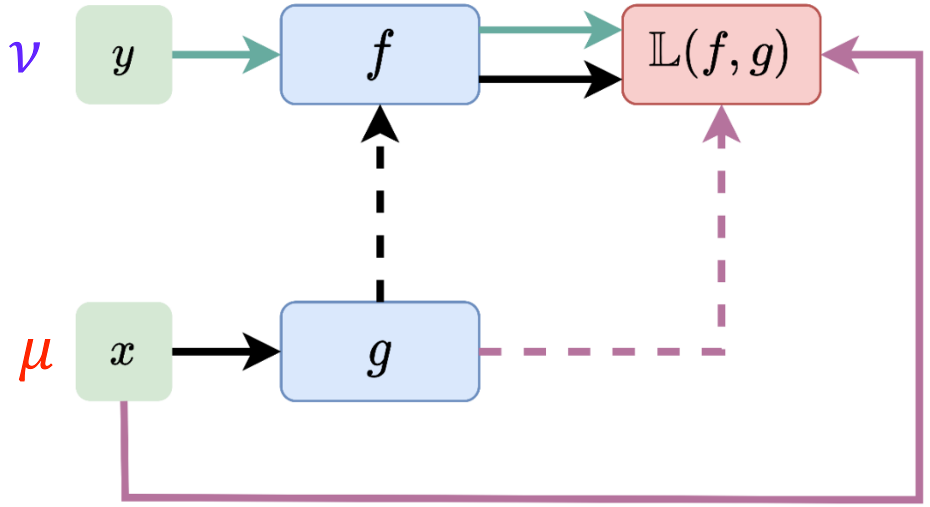
Optimal transport can be solved using two convex NN:

- Convex neural networks: $h(x;\theta)$ is a convex function
- Two convex neural networks - f & g
- Transport map: $\hat{T} = \nabla_x g(x;\theta')$





Optimal transport architecture

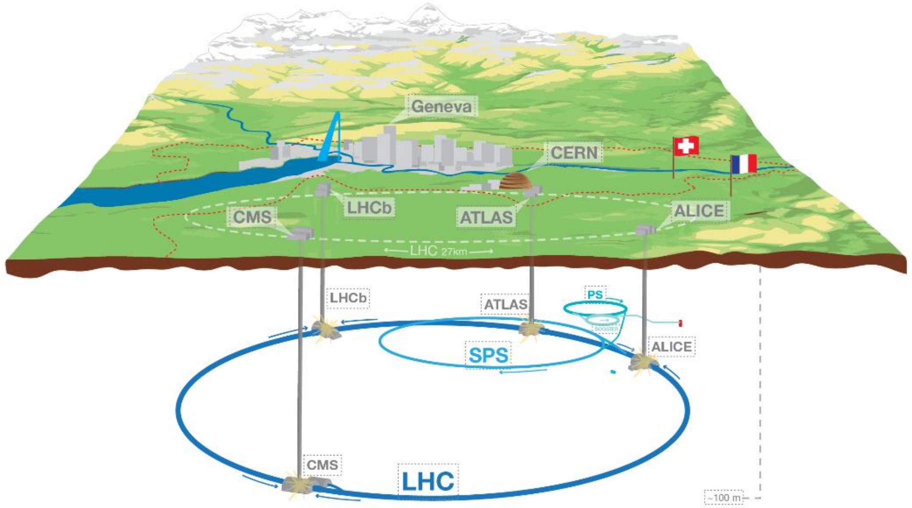


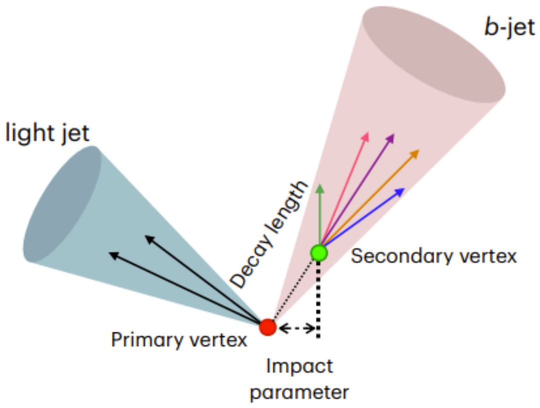
$$\mathbb{L}(\boldsymbol{\phi},\boldsymbol{\psi}) = \boldsymbol{min_f max_g} \sum \boldsymbol{f_\phi(y;\theta)} + \boldsymbol{x \cdot \nabla_x g_\psi(x;\theta')} - \boldsymbol{f_\phi(\nabla_x g_\psi(x;\theta'),\theta')}$$ **with the** $\hat{\boldsymbol{T}} = \boldsymbol{\nabla_x g(x;\theta')}$

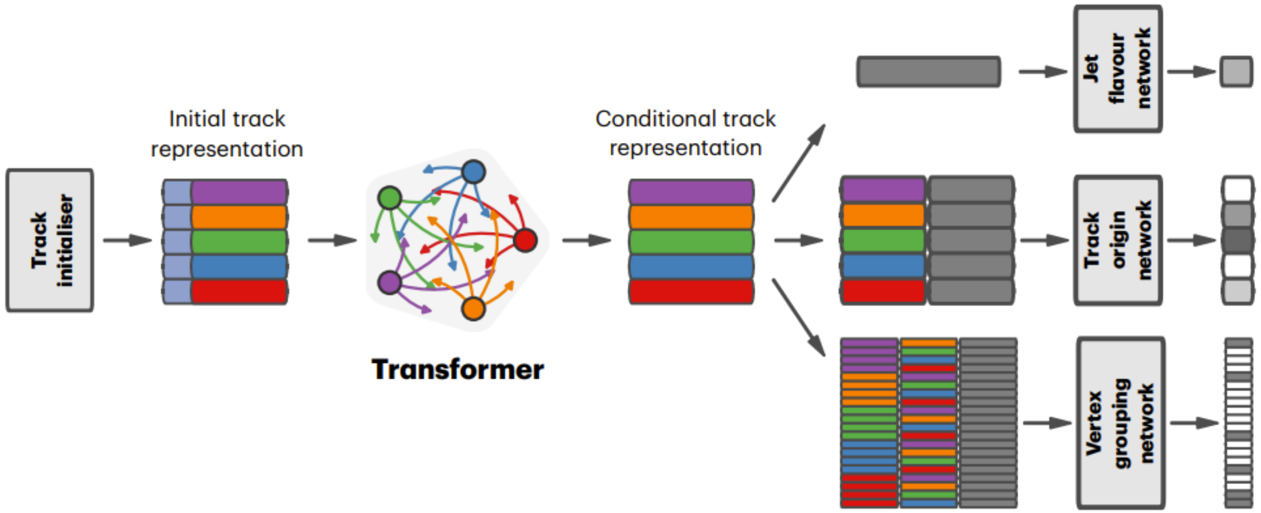
Optimal transport in HEP

Flavour tagging in ATLAS:

1. Classify the flavour of a hadronic decay
   A. Using transformer classifier using CE
   B. Trained on simulation (mismodelled)
   C. Evaluated and compare on real data (Domian shift)



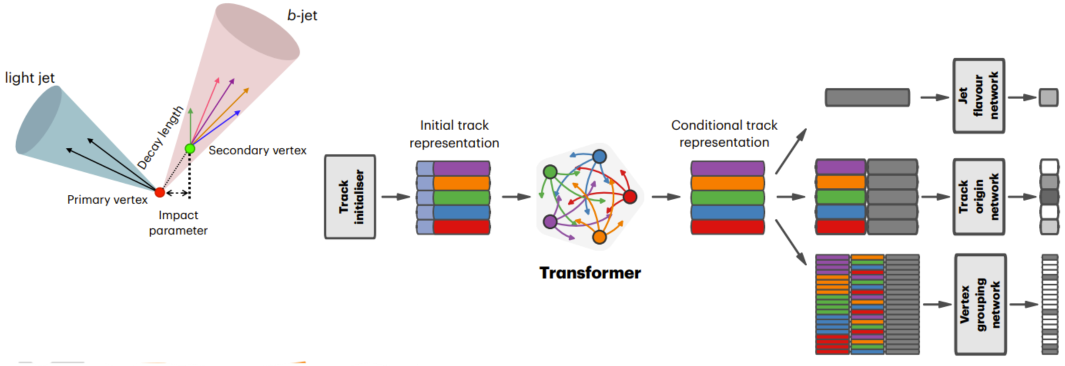


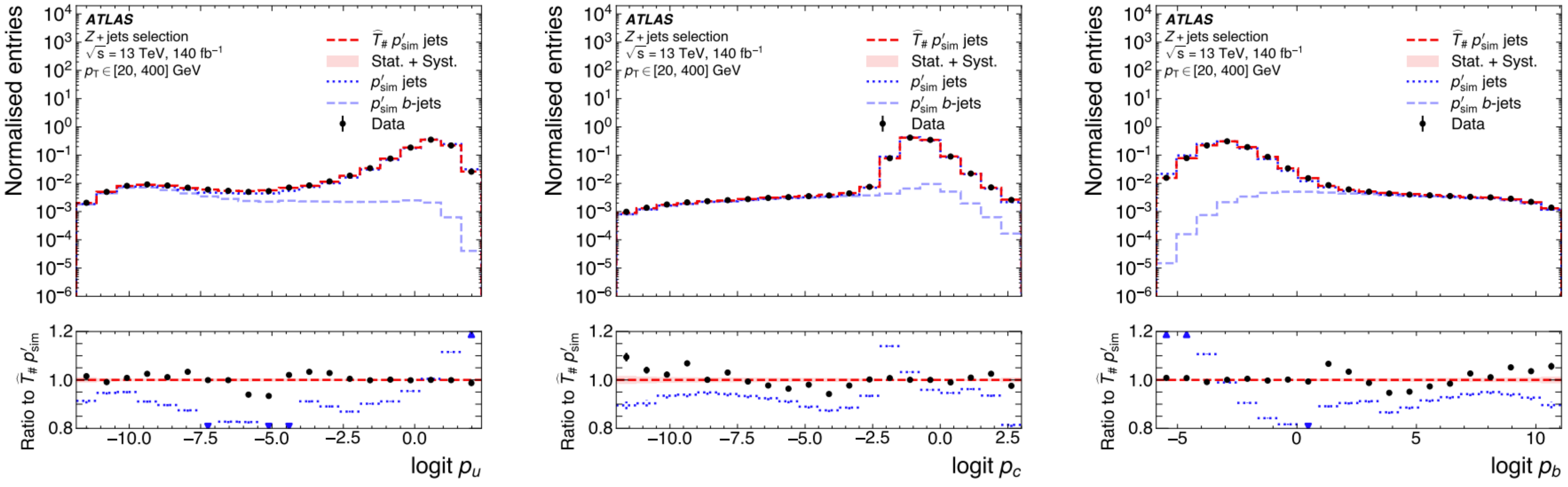


Optimal transport in HEP

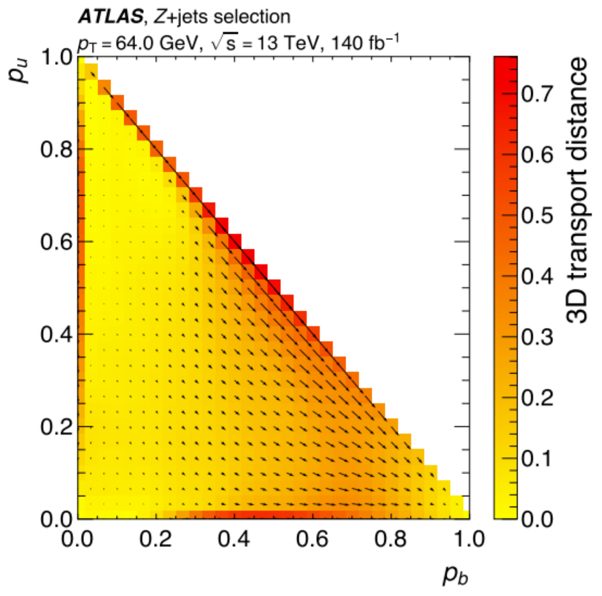
Calibrate flavour tagging in ATLAS:

1. Evaluated on real data (Domian shift)
   A. Find the optimal transport (simulate > data)
   B. Physics measurement on correct/calibrated simulation
      - Simulation adapted to the data





(a) (b) (c)

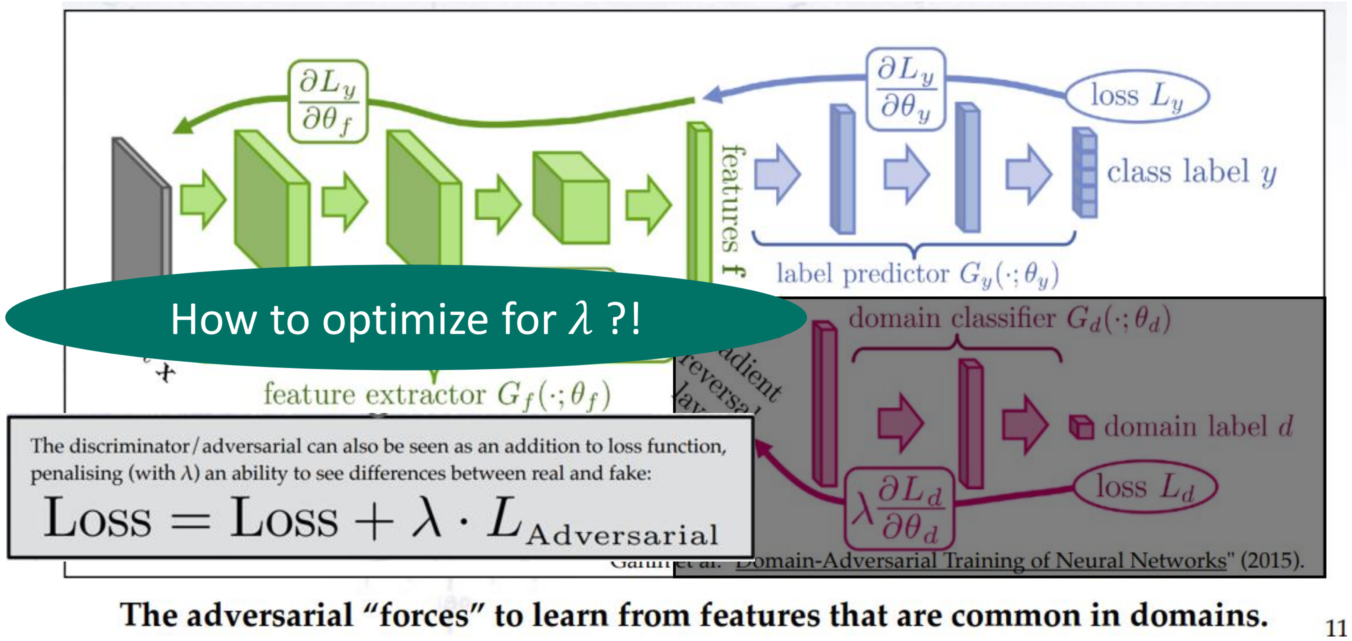


(b)

Optimal transport in HEP

Decorrelate against protected variables (gender, race):

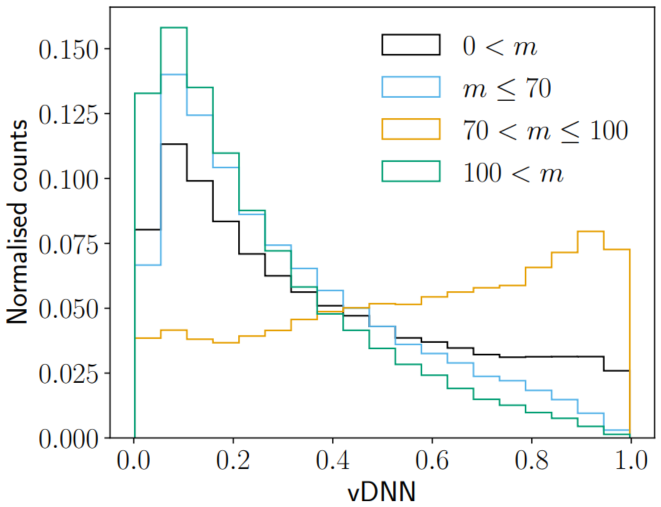
1. Train a classifier to give bank loans
2. The classifier should not use your weight as a feature
   - Do not have direct access to your weight
   - It can see if you subscript to a gym



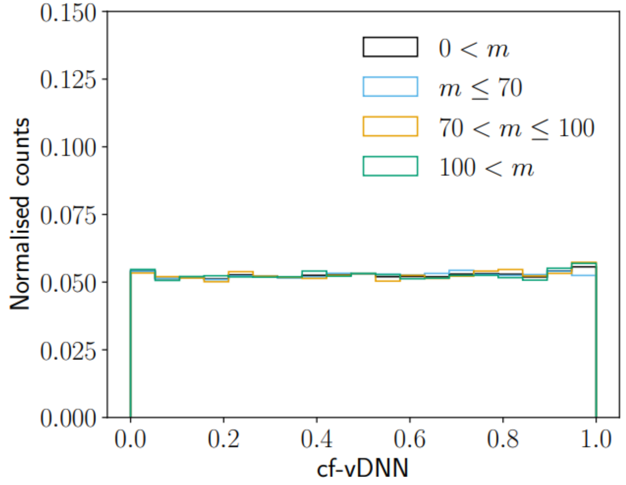
Give loan

Predict mass m

Biased classification



Unbiased classification

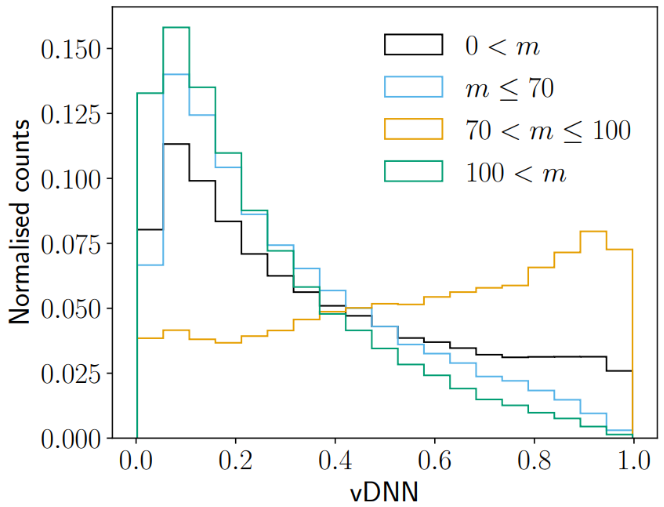


Optimal transport in HEP

Decorrelate against protected variables (gender, race):

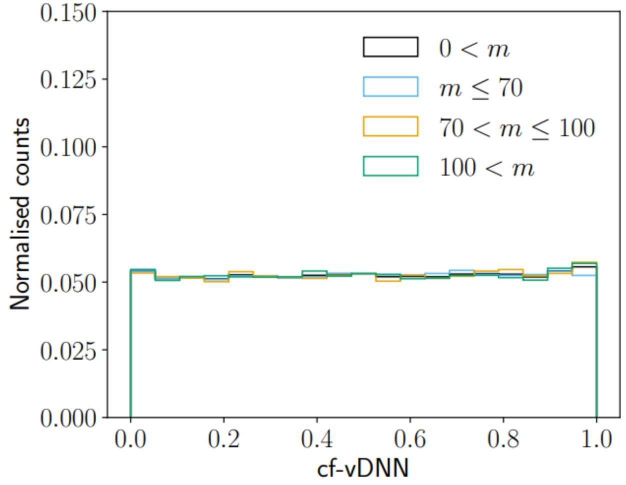
1. Train a classifier to give bank loans without any decorrelation
2. Find the optimal transport between $T_m\big(p(x|m)\big) = p(x)$
    - A. Work as a post-processing of the discriminate score
    - B. More stable than decorrelation during training
    - C. Scales better to higher dimensions

Biased classification



$$T_m\big(p(x|m)\big) = p(x)$$

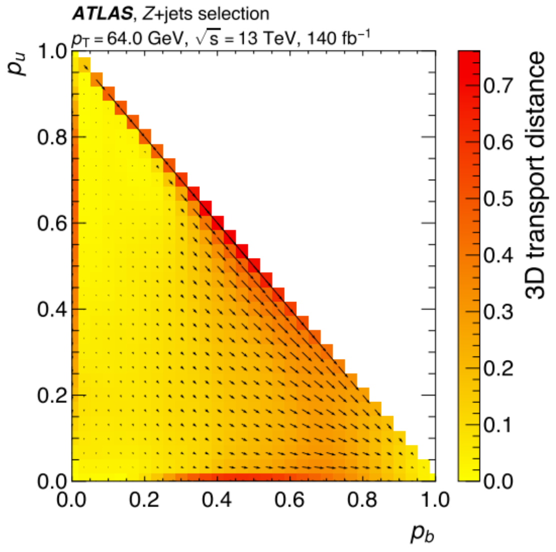
Unbiased classification



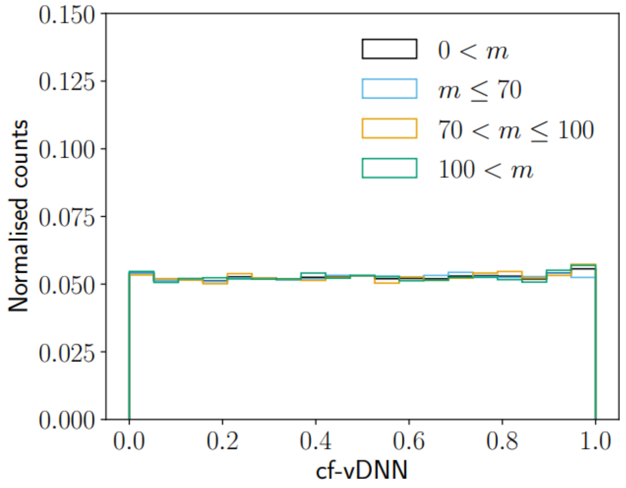
Outlook

Optimal transport - very active field of research:

- Many algorithms in development (sinkhorn, neural solver etc.)
- Various application in science
- Looking for the best ways to solve OT
- High energy physics is the perfect playground
    - All forms of data sample



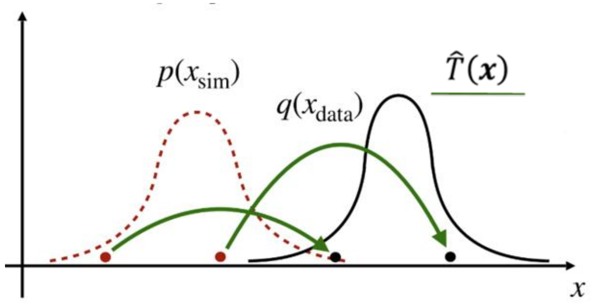
(b)



Thank you for listening!
Questions?

Backup slides

Optimal transport



Quick introduction to Optimal Transport and the Kantorovich duality

- Finding the optimal transport map $\hat{T}(x)$ that satisfies:

$$\hat{T}(x) = \underset{T(x):p'(x')\equiv q(x)}{\operatorname{arginf}} \int dx\, p(x)\, c[x, T(x)]$$

- It can be formulated into a dual optimisation problem

$$\hat{f}(y;\theta) = \underset{f\in cvx(T)}{\operatorname{arginf}} \int dy\, q(y) f(y;\theta) + \int dx\;\, p(x|\theta) f^*(x;\theta) \text{ with } f^*(x;\theta) = \sup_{y\in Y} x\cdot y - f(y;\theta)$$

- Optimising over $f^*(x;\theta)$ and substituting it with another *convex* function

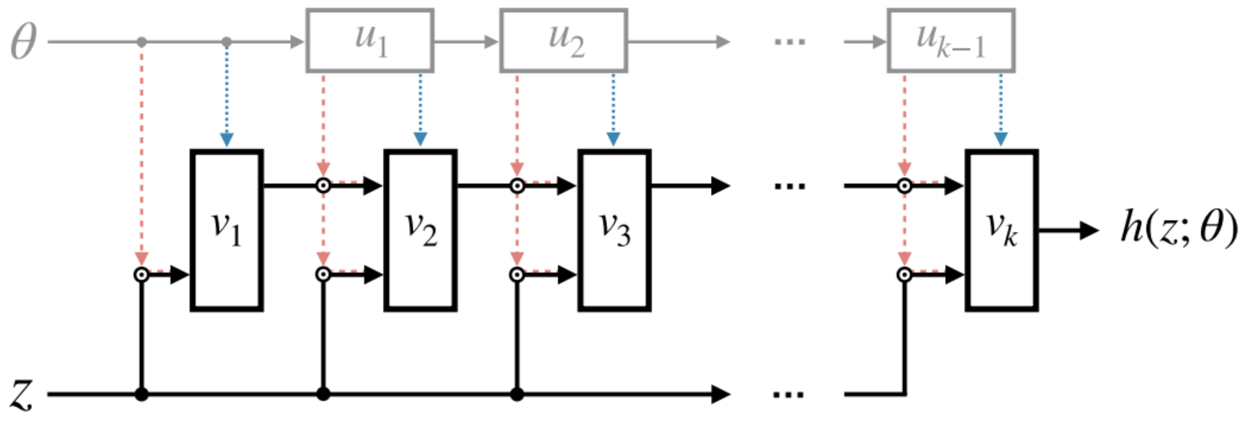
$$\mathbb{L}(\boldsymbol{\phi},\boldsymbol{\psi}) = \mathbf{min}_f \mathbf{max}_g \sum \boldsymbol{f}_{\boldsymbol{\phi}}(\boldsymbol{y};\boldsymbol{\theta}) + \boldsymbol{x}\cdot\nabla_x \boldsymbol{g}_{\boldsymbol{\psi}}(\boldsymbol{x};\boldsymbol{\theta}') - \boldsymbol{f}_{\boldsymbol{\phi}}(\nabla_x \boldsymbol{g}_{\boldsymbol{\psi}}(\boldsymbol{x};\boldsymbol{\theta}'),\boldsymbol{\theta}') \textbf{ with the } \widehat{\boldsymbol{T}} = \nabla_x \boldsymbol{g}(\boldsymbol{x};\boldsymbol{\theta}')$$

Which can be minimised using two convex networks f and g

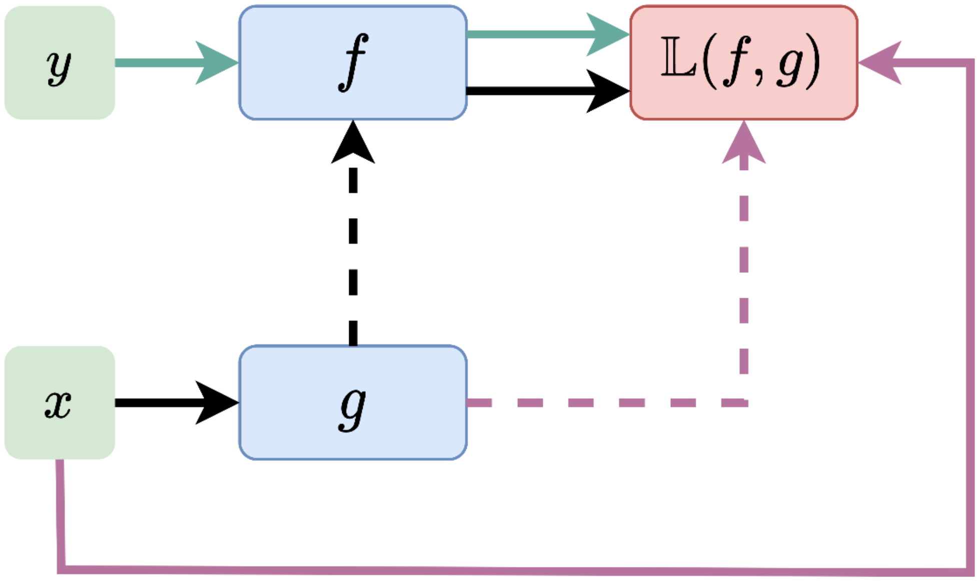
Optimal Neural Solver

Optimal transport can be solved using two convex NN:

- f & g has to be convex (see figure below)
- The conditional distributions θ & θ' are required to have the same PDF



Optimal transport architecture



x/y can also contain conditions θ/θ'

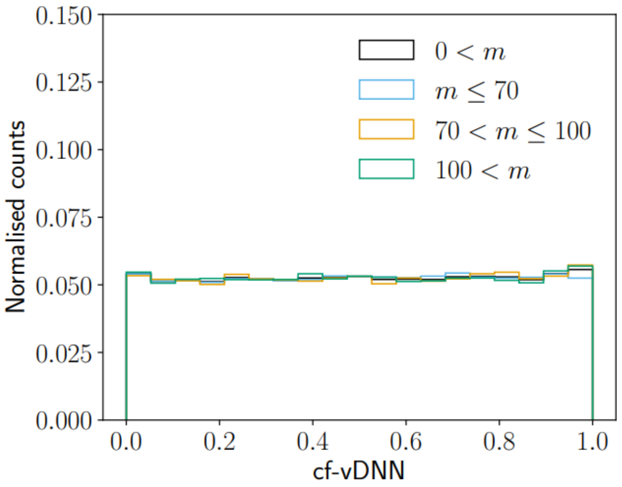
$$\mathbb{L}(\phi,\psi) = \min_f \max_g \sum f_\phi(y;\theta) + x \cdot \nabla_x g_\psi(x;\theta') - f_\phi(\nabla_x g_\psi(x;\theta'),\theta') \text{ with the } \hat{T} = \nabla_x g(x;\theta')$$

Optimal transport in HEP

Decorrelation to protected variables (gender, sex, race)

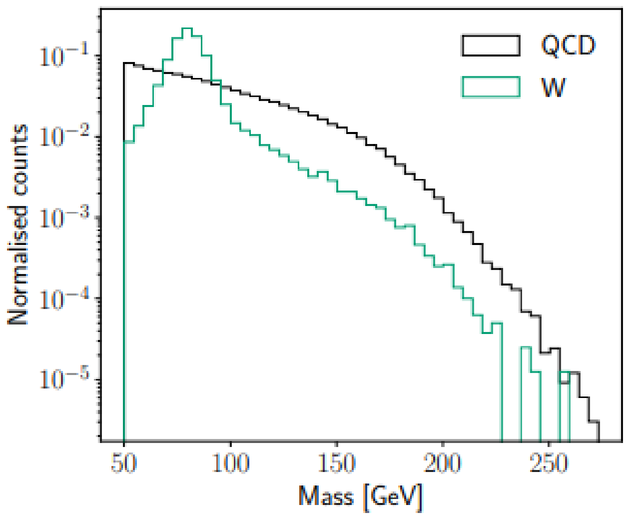
- Classifier correlated to mass
- This can be transformed to a space independent on mass

Unbiased classification



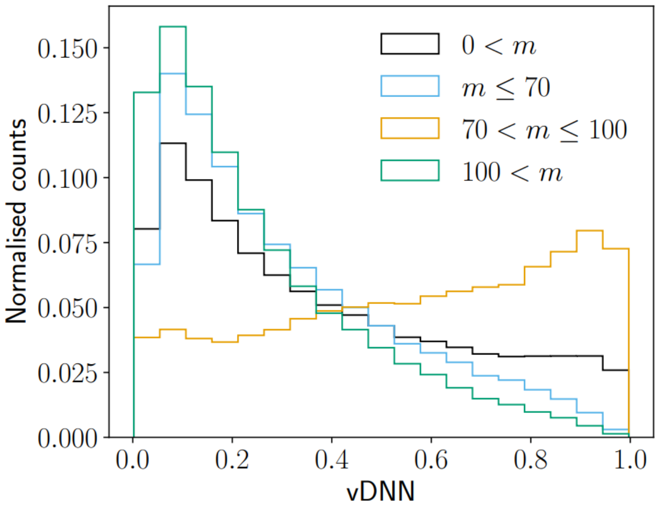
Decorrelate

Biased dataset



Results in

Biased classification



[Decorrelation with Conditional Normalizing Flows]