Learning Representations using Neural Networks and Optimal Transport

Warith Harchaoui

Ph.D. Defense

October 8, 2020

Learning Representations using
Neural Networks and
Optimal Transport

1. Introduction

2. Clustering

3. Prediction with Uncertainty

4. Unsupervised Features Importance

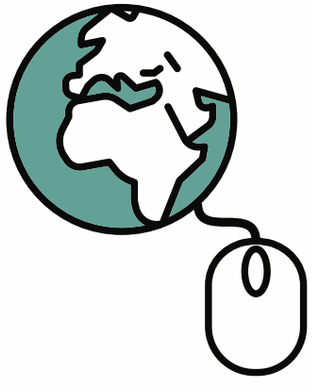
5. Conclusion

Clustering

Marketing

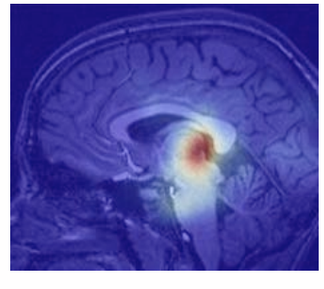
Genomics

Documents Analysis



Prediction with Uncertainty

Medical Diagnostics



Industrial Annotation

Robotic Exploration

Unsupervised Features Importance

Data Understanding

High Dimensionality Analysis

Information Retrieval

Representations

Clustering

Prediction with Uncertainty

Features Importance

All models are wrong, but some are useful

George Box, Science and Statistics, 1976

Clustering Outline

Clustering is ill-posed

Auto-Encoders

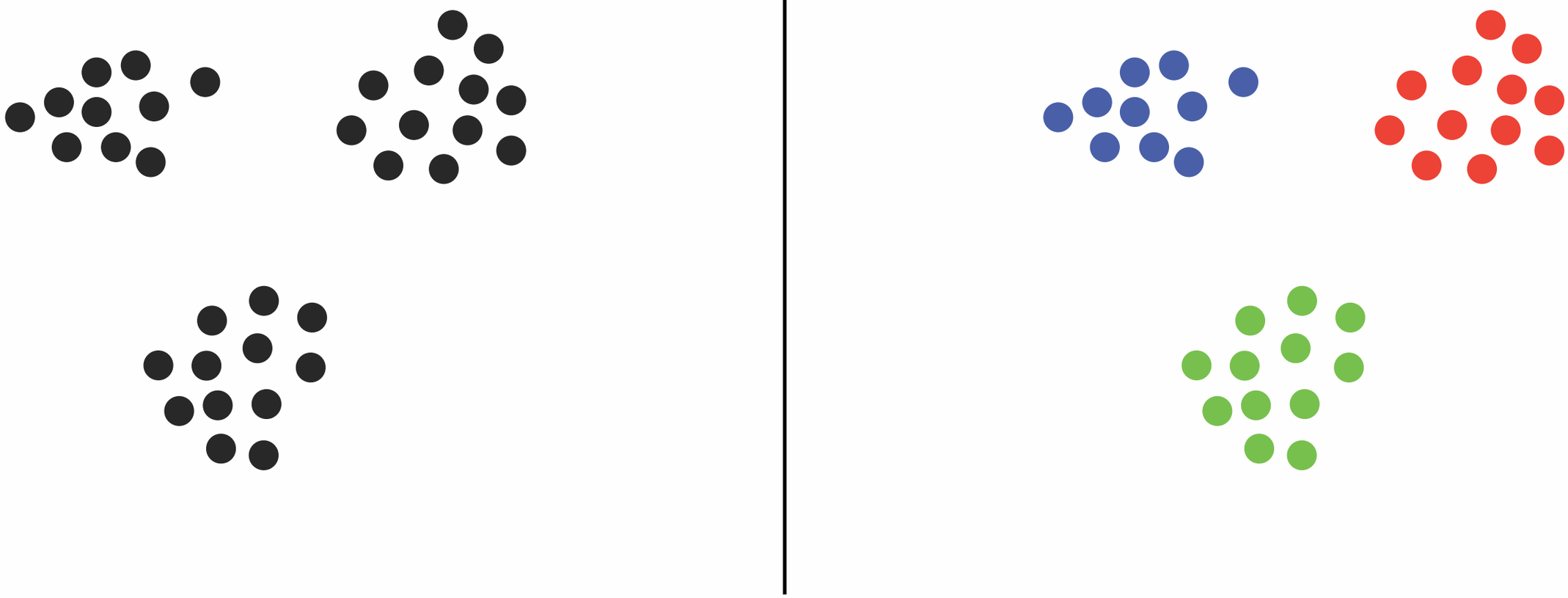
Optimal Transport

Wasserstein Generative Adversarial Networks
Wasserstein GAN

Generative Wasserstein Clustering
GeWaC

Discriminative Wasserstein Clustering
DiWaC

Clustering



Clustering is ill-posed

Impossibility theorem for Clustering
Kleinberg, 2002

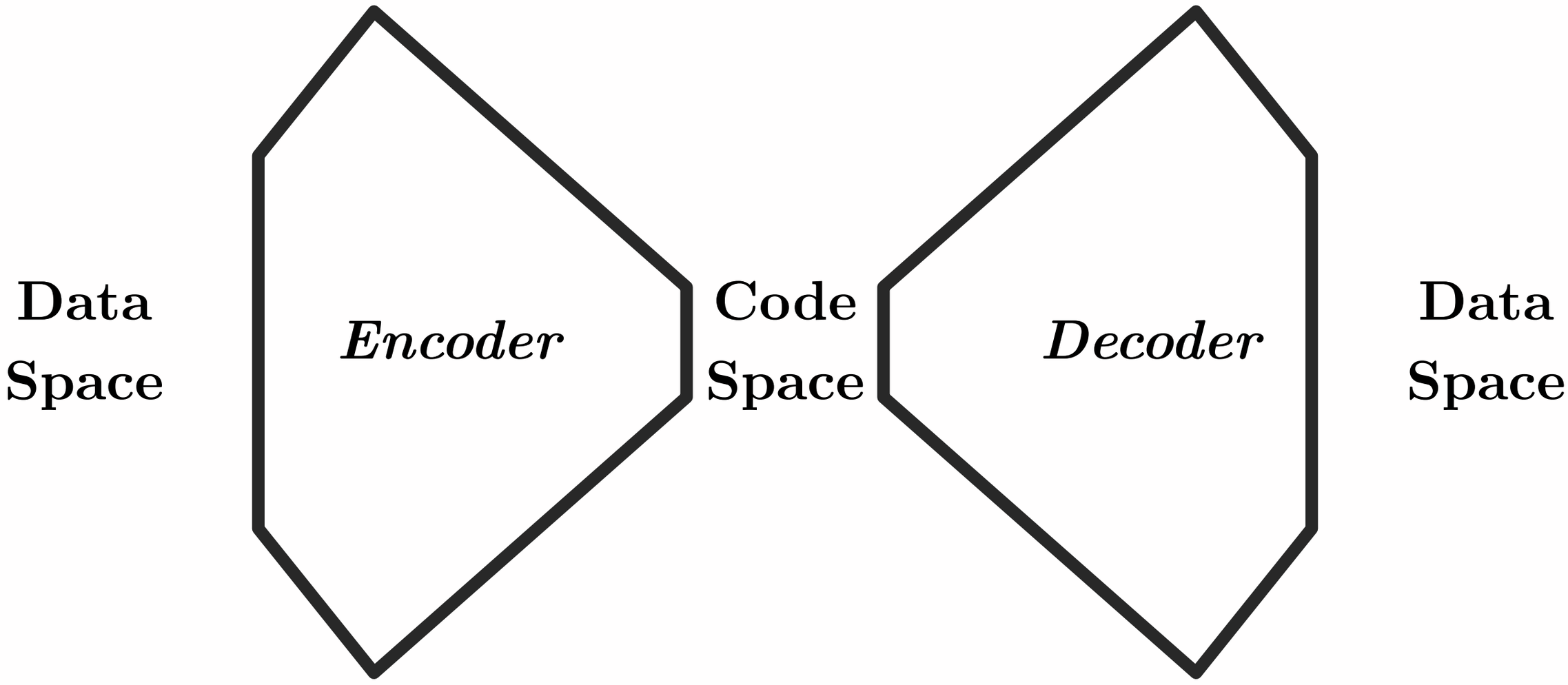
No Clustering algorithm can simultaneously verify these 3 properties

Scale Invariance *e.g.* neighbourhood threshold fails

Cluster Shapes Invariance *e.g.* k-Means fails on the Moons data

Metric Invariance *e.g.* broken pairwise relationships

Auto-Encoders



Cluster Shapes Invariance

Optimal Transport

A

B

Cost of moving A *to* B

Generative Adversarial Networks

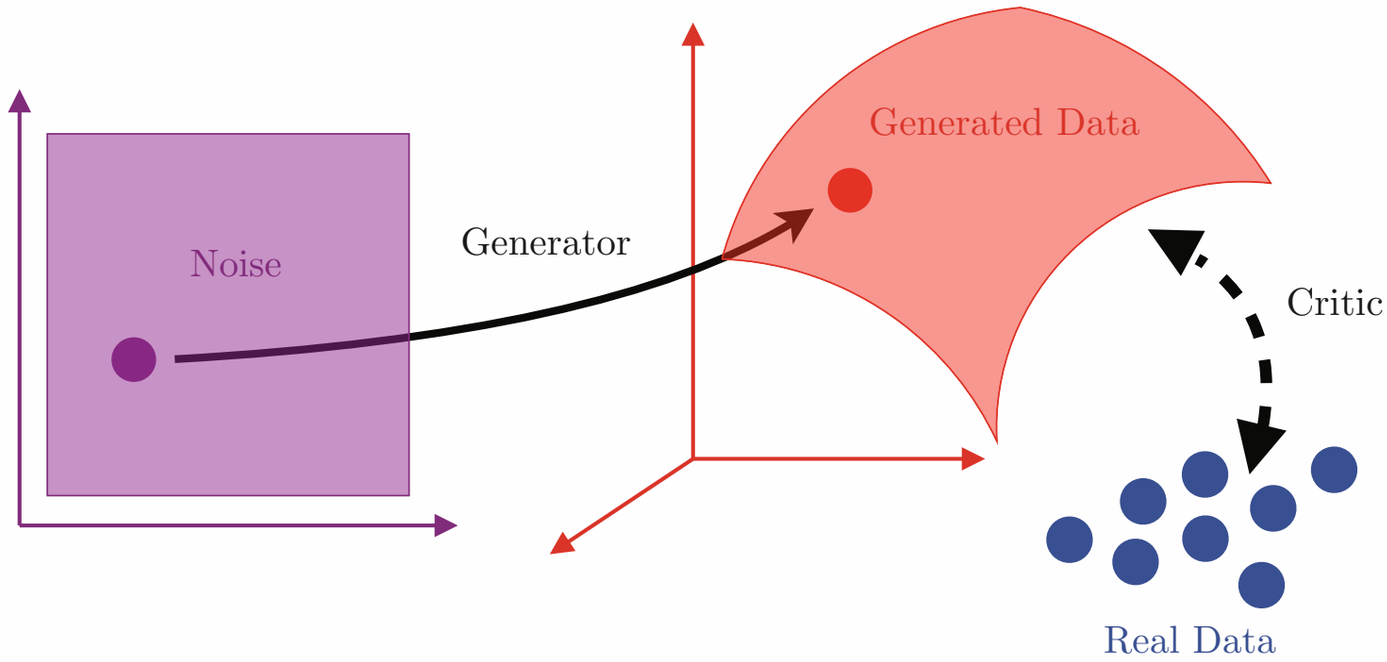


Image adapted from

Computational Optimal Transport, Peyré and Cuturi, 2018

Wasserstein GAN

Kantorovich-Rubinstein Formulation $(L_2 \text{ cost: } c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2)$

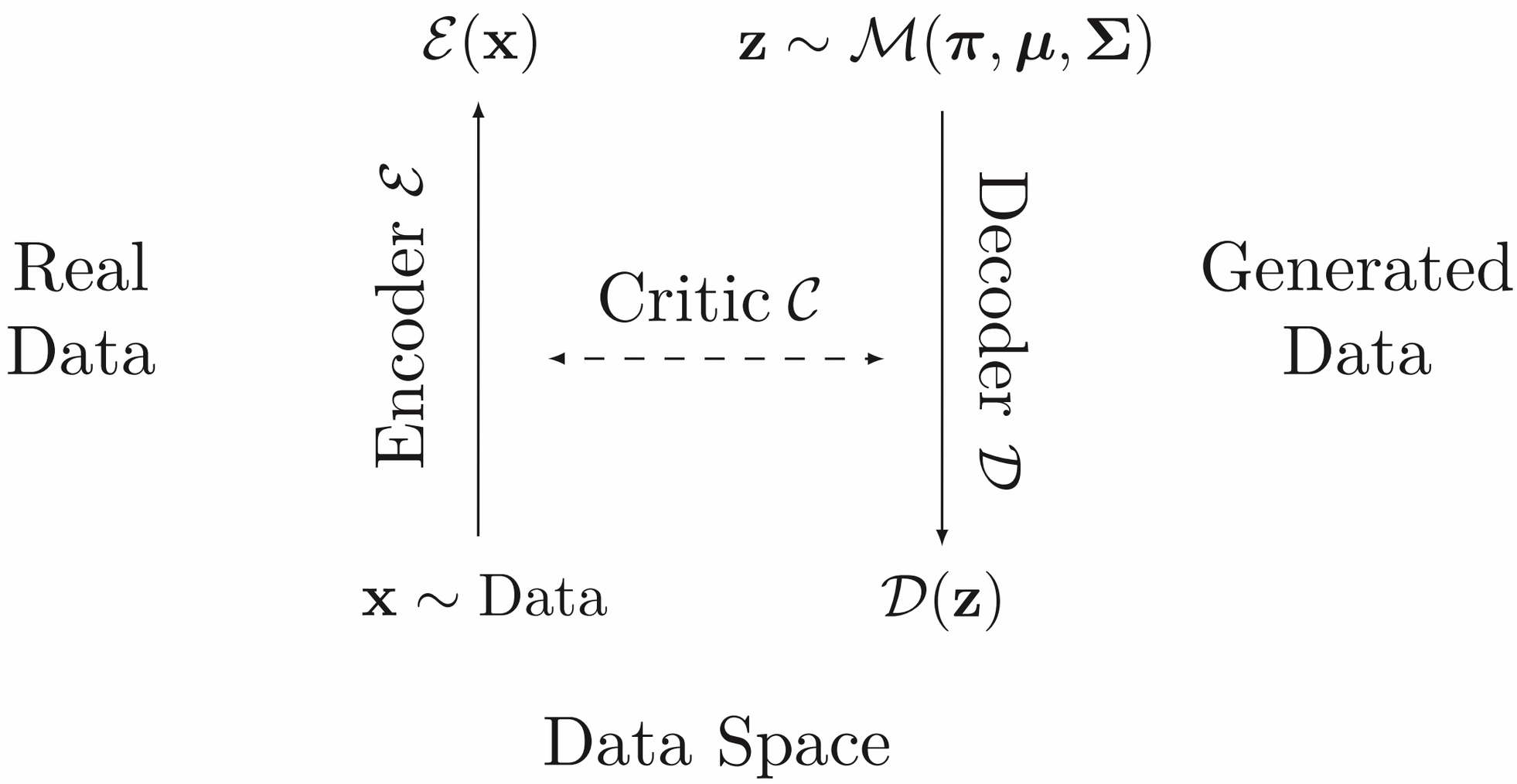
$$W_{\boldsymbol{c}}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \sup_{\mathcal{C} \in \text{Lip-1}} \mathbb{E}_{\mathbf{x} \sim \boldsymbol{\mu}} [\mathcal{C}(\mathbf{x})] - \mathbb{E}_{\mathbf{y} \sim \boldsymbol{\nu}} [\mathcal{C}(\mathbf{y})]$$

where Lip-1 is the 1-Lipschitz functions set (from $\mathcal{X} \subset \mathbb{R}^D$ to $\mathbb{R}$).

$$\min_{\mathcal{G}} \max_{\mathcal{C} \in \text{Lip}_1} \mathbb{E}_{\mathbf{x} \sim p} \left[\mathcal{C}\left(\mathbf{x}\right) \right] - \mathbb{E}_{\boldsymbol{\epsilon}} \left[\mathcal{C}\left(\mathcal{G}(\boldsymbol{\epsilon})\right) \right]$$

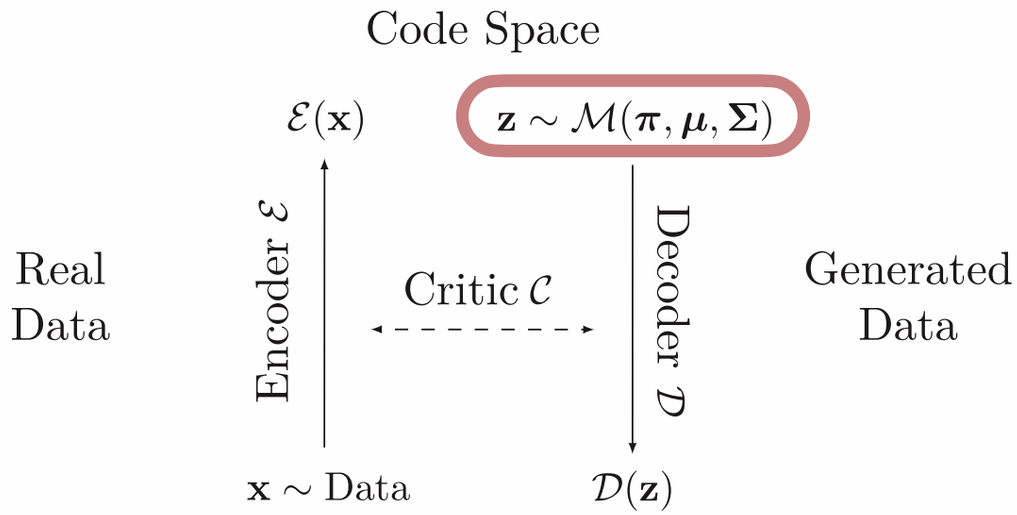
Generative Wasserstein Clustering

GeWaC



Generative Wasserstein Clustering

GeWaC



Data Space

Reparametrization Trick

(borrowed from Kingma et Welling [2014]
and well explained in Shakir Mohamed et al. [2020])

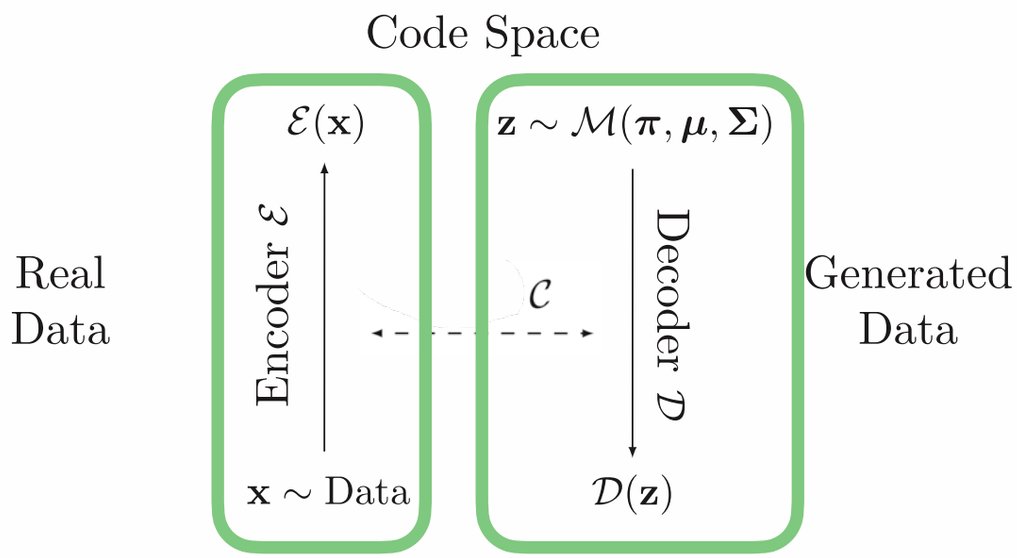
$$\mathbf{z} \sim \mathcal{M}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\mathbb{E}_{\mathbf{z} \sim \mathcal{M}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}[g(\mathbf{z})] = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)} \left[g\left(\boldsymbol{\mu}_k + \mathbf{C}_k \times \boldsymbol{\epsilon} \right) \right]$$

$$\text{(where } \mathbf{C}_k \times \mathbf{C}_k^\top = \boldsymbol{\Sigma}_k \text{)}$$

Generative Wasserstein Clustering

GeWaC

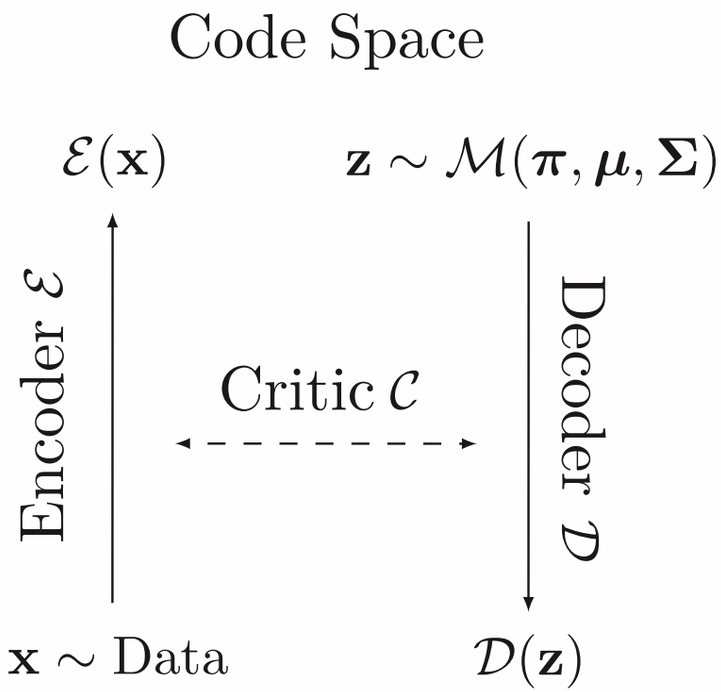


Concatenation Trick

(borrowed from Dumoulin et al. [2017]
with reciprocity proof from Donahue et al. [2016])

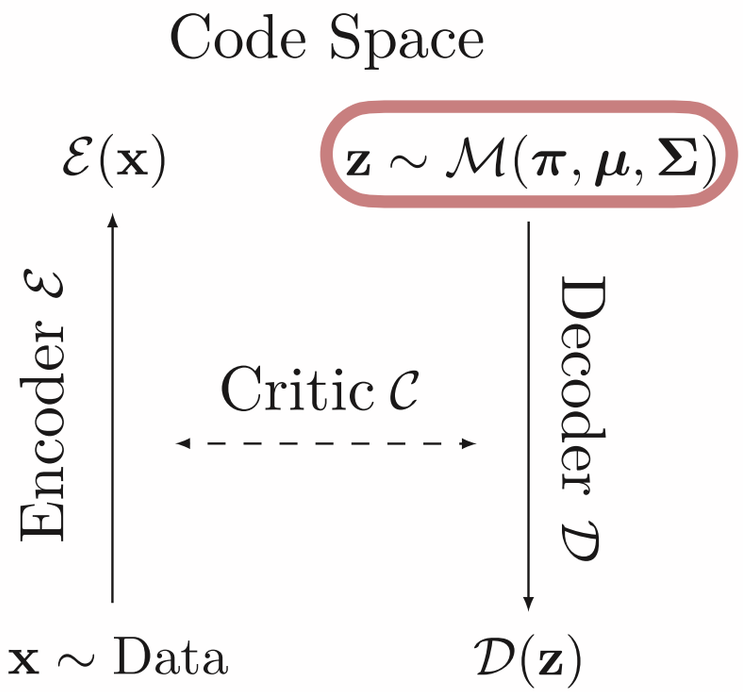
- $\tilde{\mathbf{x}} = a(\mathbf{x}) = \left[\mathbf{x}^\top, \mathcal{E}(\mathbf{x})^\top\right]^\top \sim \tilde{p}$ considered as *real* with $\mathbf{x}$ representing data
- $\tilde{\mathbf{y}} = b(\mathbf{z}) = \left[\mathcal{D}(\mathbf{z})^\top, \mathbf{z}^\top\right]^\top \sim \tilde{q}$ considered as *generated* with $\mathbf{z}$ sampled over a parametrized mixture $q = \mathcal{M}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Generative Wasserstein Clustering GeWaC



Generative Wasserstein Clustering

GeWaC

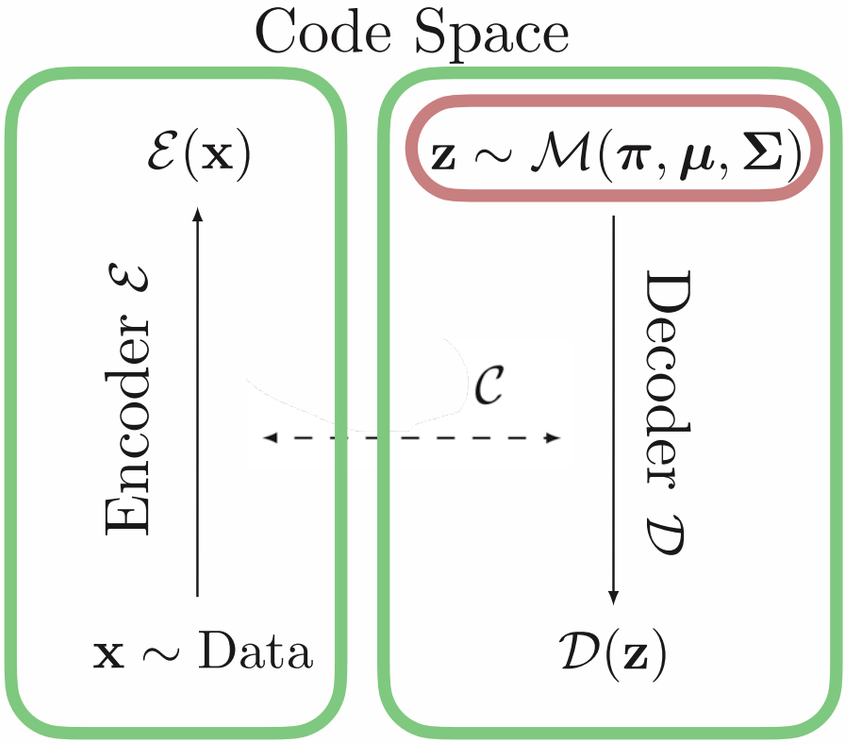


Reparametrization

$$\tilde{\mathbf{z}} \sim \mathcal{M}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Generative Wasserstein Clustering

GeWaC



Reparametrization

$$\tilde{\mathbf{z}} \sim \mathcal{M}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

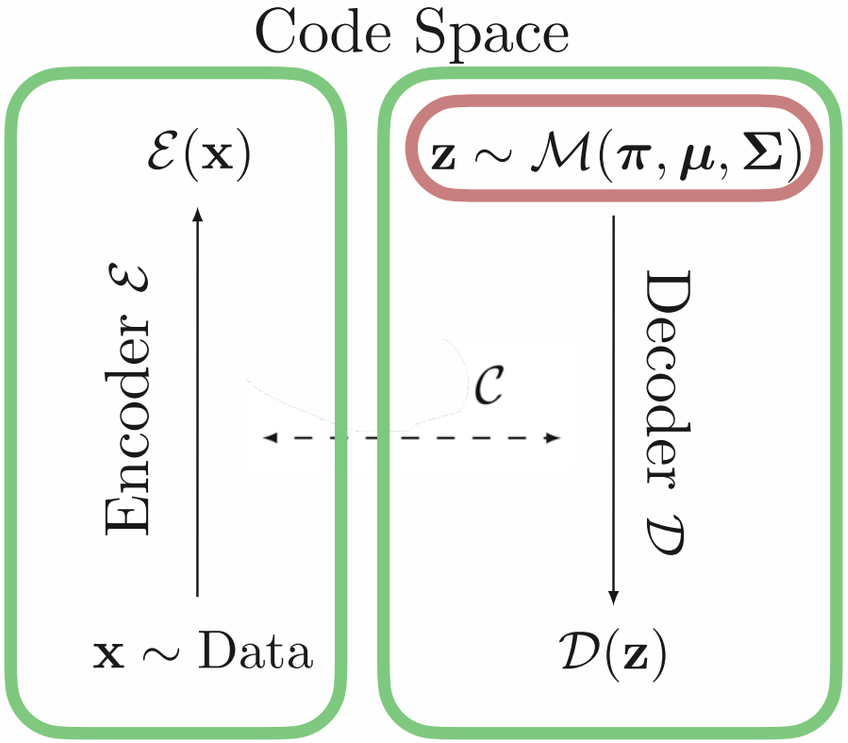
$$\tilde{\mathbf{y}} = [\mathcal{D}(\mathbf{z})^\top, \mathbf{z}^\top]^\top \sim \tilde{q}$$

$$\tilde{\mathbf{x}} = [\mathbf{x}^\top, \mathcal{E}(\mathbf{x})^\top]^\top \sim \tilde{p}$$

Concatenation

Generative Wasserstein Clustering

GeWaC



Reparametrization

$$\tilde{\mathbf{z}} \sim \mathcal{M}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\tilde{\mathbf{y}} = [\mathcal{D}(\mathbf{z})^\top, \mathbf{z}^\top]^\top \sim \tilde{q}$$

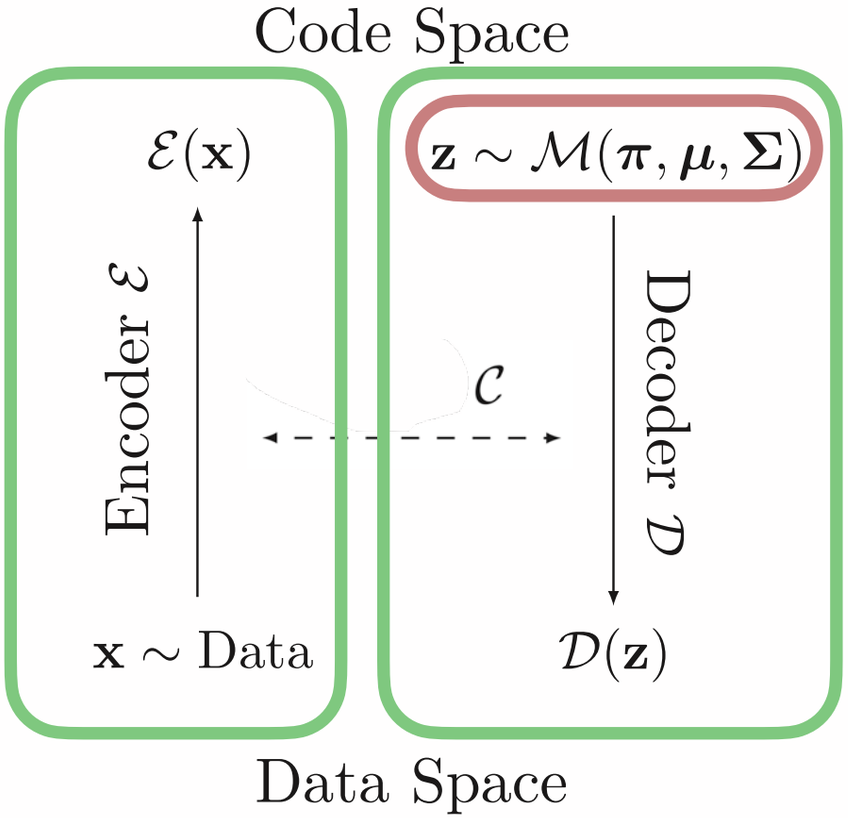
$$\tilde{\mathbf{x}} = [\mathbf{x}^\top, \mathcal{E}(\mathbf{x})^\top]^\top \sim \tilde{p}$$

Concatenation

$$\min_{\mathcal{E},\mathcal{D},\mathcal{M}} W(\tilde{q}, \tilde{p}) \quad \text{and} \quad W(\tilde{q}, \tilde{p}) = \max_{\mathcal{C} \in \mathrm{Lip}_1} \mathbb{E}_{\tilde{\mathbf{y}} \sim \tilde{q}}[\mathcal{C}(\tilde{\mathbf{y}})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim \tilde{p}}[\mathcal{C}(\tilde{\mathbf{x}})]$$

Generative Wasserstein Clustering

GeWaC



Reparametrization

$$\tilde{\mathbf{z}} \sim \mathcal{M}(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\tilde{\mathbf{y}} = [\mathcal{D}(\mathbf{z})^\top, \mathbf{z}^\top]^\top \sim \tilde{q}$$

Concatenation

$$\tilde{\mathbf{x}} = [\mathbf{x}^\top, \mathcal{E}(\mathbf{x})^\top]^\top \sim \tilde{p}$$

$$\min_{\mathcal{E},\mathcal{D},\mathcal{M}} W(\tilde{q}, \tilde{p}) \quad \text{and} \quad W(\tilde{q}, \tilde{p}) = \max_{\mathcal{C} \in \mathrm{Lip}_1} \mathbb{E}_{\tilde{\mathbf{y}} \sim \tilde{q}}[\mathcal{C}(\tilde{\mathbf{y}})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim \tilde{p}}[\mathcal{C}(\tilde{\mathbf{x}})]$$

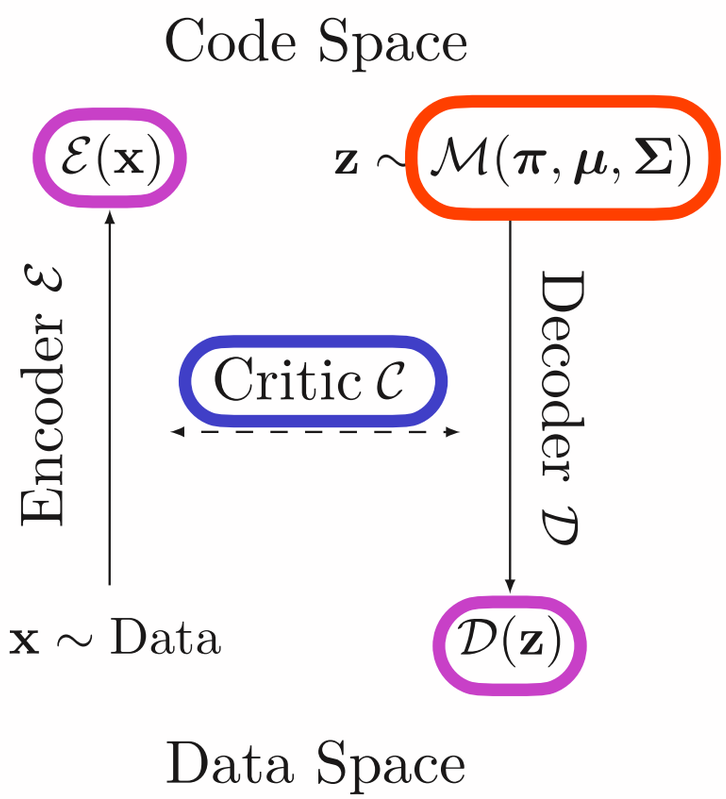
$$\min_{\mathcal{E},\mathcal{D},\mathcal{M}} \max_{\mathcal{C} \in \mathrm{Lip}_1} \mathbb{E}_{\tilde{\mathbf{y}} \sim \tilde{q}}[\mathcal{C}(\tilde{\mathbf{y}})] - \mathbb{E}_{\tilde{\mathbf{x}} \sim \tilde{p}}[\mathcal{C}(\tilde{\mathbf{x}})]$$

Generative Wasserstein Clustering

GeWaC

In practice, we do a three-steps initialization for our optimization

1. Vanilla Auto-encoder
2. Gaussian Mixture Model on the Codes
3. Critic Warm-Start
4. Final Optimization



Generative Wasserstein Clustering GeWaC

$$\min_{\mathcal{E},\mathcal{D},\mathcal{M}} \max_{\mathcal{C}\in \mathrm{Lip}_1} \mathbb{E}_{\tilde{\mathbf{y}}\sim\tilde{q}}[\mathcal{C}(\tilde{\mathbf{y}})] - \mathbb{E}_{\tilde{\mathbf{x}}\sim\tilde{p}}[\mathcal{C}(\tilde{\mathbf{x}})]$$

$$\mathbb{E}_{\tilde{\mathbf{y}}\sim\tilde{q}}[\mathcal{C}(\tilde{\mathbf{y}})] = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0}_d,\mathbf{I}_d)} \left[\mathcal{C}\left(\left[\mathcal{D}(\mathbf{S}_k \times \boldsymbol{\epsilon} + \boldsymbol{\mu}_k)^\top, (\mathbf{S}_k \times \boldsymbol{\epsilon} + \boldsymbol{\mu}_k)^\top \right]^\top \right) \right]$$

Instable Proportions ☹

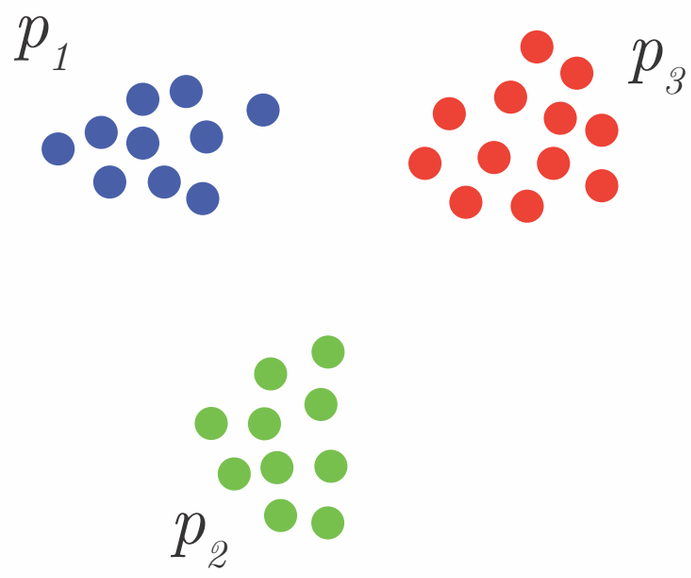
Discriminative Wasserstein Clustering

DiWaC

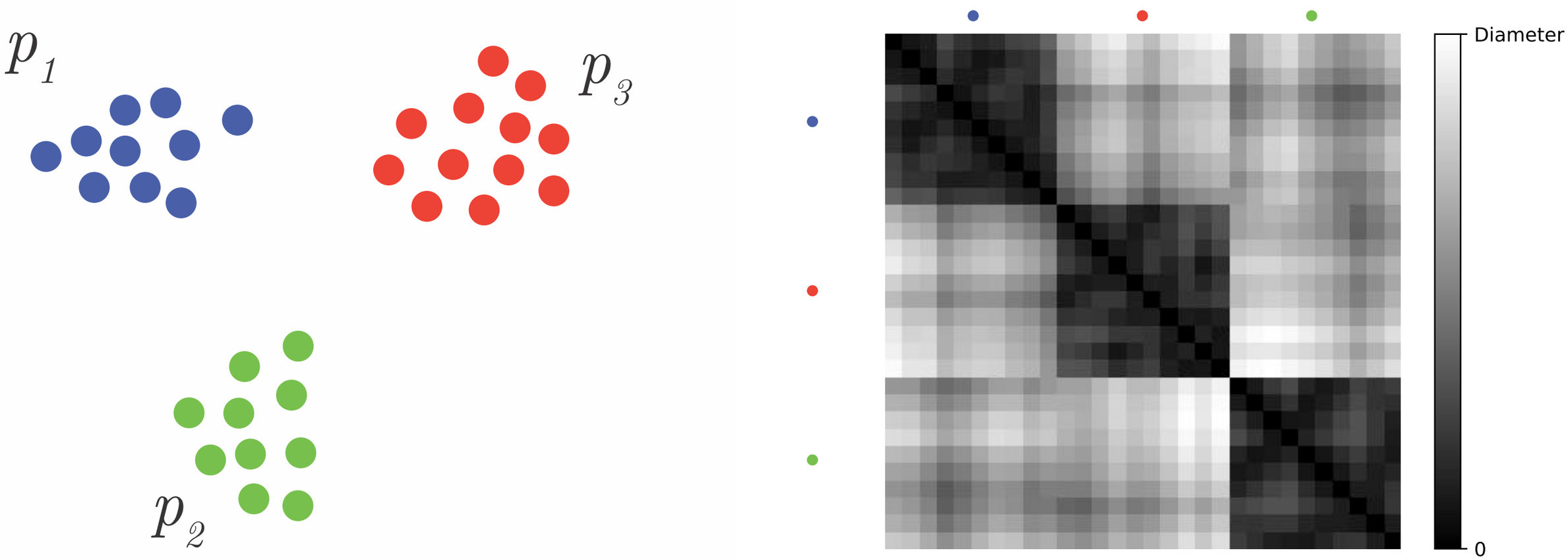
$$\mathbf{x} \sim p = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times p_k$$

Instead of bringing distributions together,

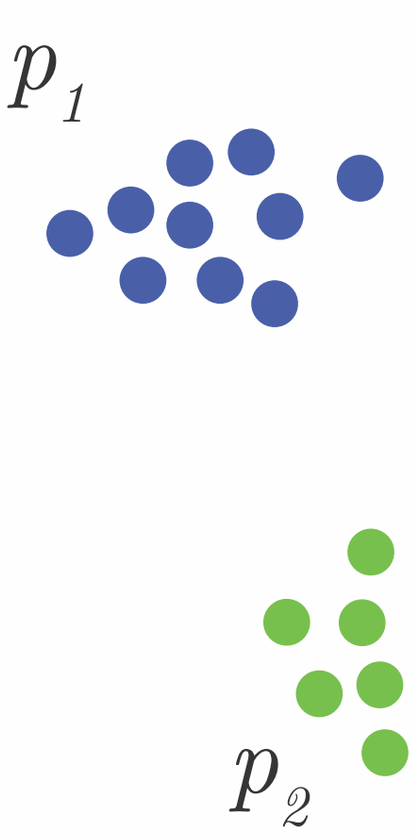
Let's bring them far from each other!

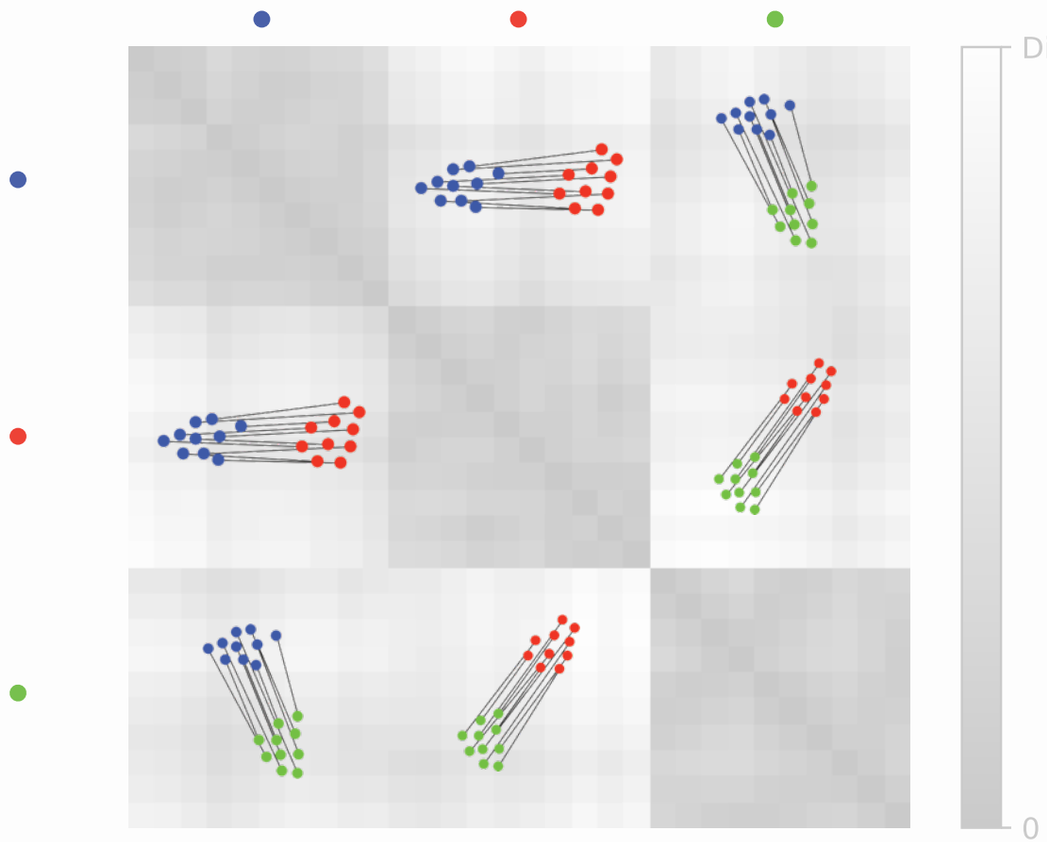


Discriminative Wasserstein Clustering DiWaC



Discriminative Wasserstein Clustering DiWaC





Discriminative Wasserstein Clustering

DiWaC

$$\mathcal{L}_{\mathrm{u}}^{\mathrm{OvO}}(p_1, \ldots, p_K) = \frac{1}{K^2} \sum_{k=1}^{K} \sum_{k'=1}^{K} W(p_k, p_{k'})$$

$$\mathcal{L}_{\mathrm{n}}^{\mathrm{OvO}}(p_1, \ldots, p_K, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_K) = \sum_{k=1}^{K} \sum_{k'=1}^{K} \boldsymbol{\pi}_k \times \boldsymbol{\pi}_{k'} \times W(p_k, p_{k'})$$

$$\mathcal{L}_{\mathrm{n}}^{\mathrm{OvR}}(p_1, \ldots, p_K, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_K) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times (1 - \boldsymbol{\pi}_k) \times W(p_k, \bar{p}_k)$$

Discriminative Wasserstein Clustering DiWaC

$$\mathcal{L}_{\mathrm{u}}^{\mathrm{OvO}}(p_1, \ldots, p_K) = \frac{1}{K^2} \sum_{k=1}^{K} \sum_{k'=1}^{K} W(p_k, p_{k'})$$

$$\mathcal{L}_{\mathrm{n}}^{\mathrm{OvO}}(p_1, \ldots, p_K, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_K) = \sum_{k=1}^{K} \sum_{k'=1}^{K} \boldsymbol{\pi}_k \times \boldsymbol{\pi}_{k'} \times W(p_k, p_{k'})$$

$$\mathcal{L}_{\mathrm{n}}^{\mathrm{OvR}}(p_1, \ldots, p_K, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_K) = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times (1 - \boldsymbol{\pi}_k) \times W(p_k, \bar{p}_k)$$

$$\mathcal{L}_{\mathrm{n}}^{\mathrm{OvR}} \leq \mathcal{L}_{\mathrm{n}}^{\mathrm{OvO}}$$

Discriminative Wasserstein Clustering

DiWaC

$$\mathbf{x} \sim p = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times p_k$$

$$p_k(\mathbf{x}) = p(\mathbf{x}) \times \frac{\boldsymbol{\tau}_k(\mathbf{x})}{\boldsymbol{\pi}_k}$$

$$\boldsymbol{\tau}(\mathbf{x}) = \left[\mathbb{P}(c=1|\mathbf{x}), \dots \mathbb{P}(c=k|\mathbf{x}), \dots, \mathbb{P}(c=K|\mathbf{x})\right]^\top$$

$$\mathcal{L}_{\mathrm{n}}^{\mathrm{OvR}}(p_1, \dots, p_K, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_K) = \mathbb{E}_{\mathbf{x}\sim p}\left[\sum_{k=1}^{K}\left(\boldsymbol{\tau}_k(\mathbf{x}) - \boldsymbol{\pi}_k\right) \times \mathcal{C}_k(\mathbf{x})\right]$$

Discriminative Wasserstein Clustering

DiWaC

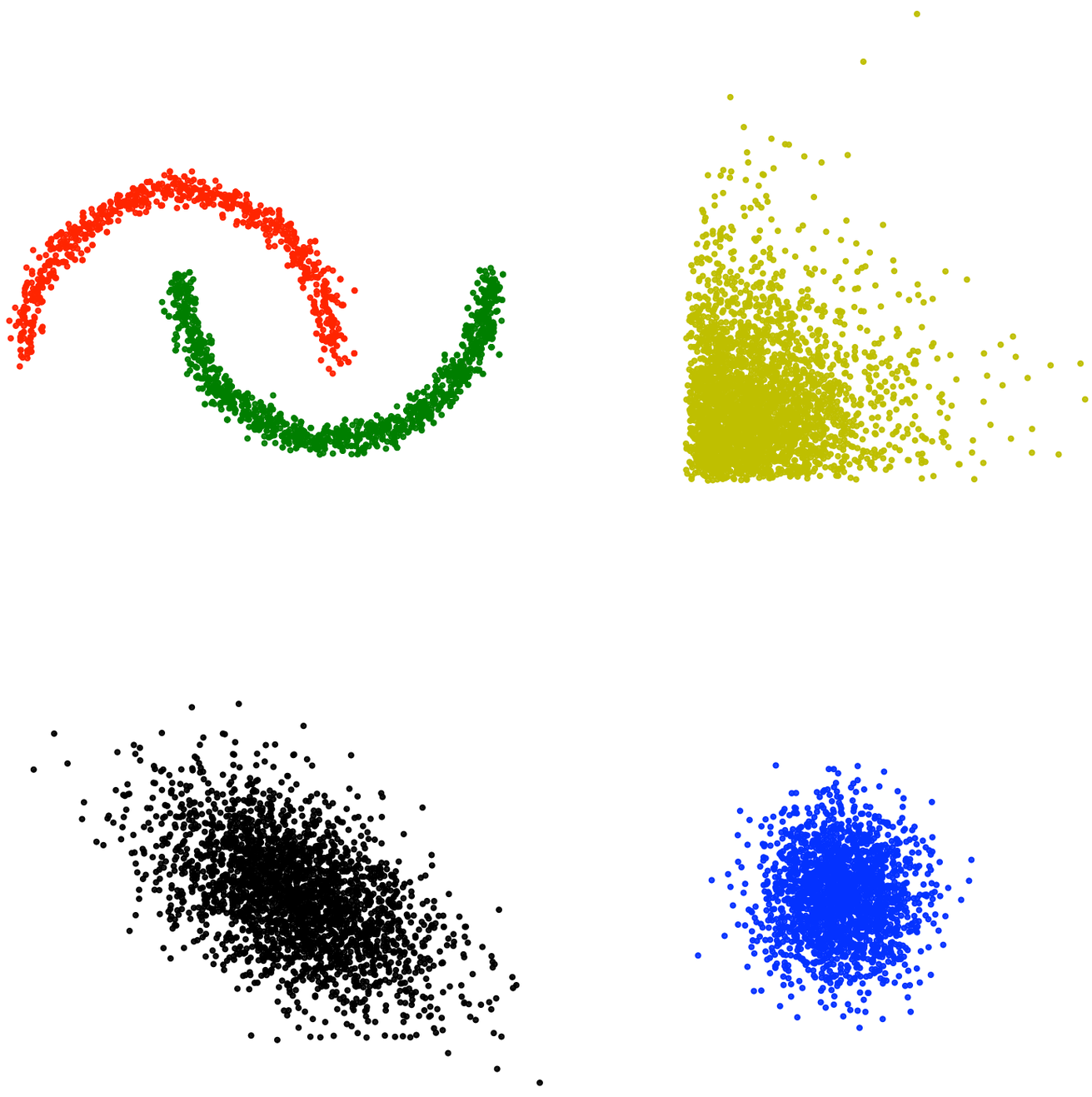
$$\mathbf{x} \sim p = \sum_{k=1}^{K} \boldsymbol{\pi}_k \times p_k$$

$$p_k(\mathbf{x}) = p(\mathbf{x}) \times \frac{\boldsymbol{\tau}_k(\mathbf{x})}{\boldsymbol{\pi}_k}$$

$$\boldsymbol{\tau}(\mathbf{x}) = \left[\mathbb{P}(c = 1|\mathbf{x}), \dots \mathbb{P}(c = k|\mathbf{x}), \dots, \mathbb{P}(c = K|\mathbf{x})\right]^\top$$

$$\mathcal{L}_{\mathrm{n}}^{\mathrm{OvR}}(p_1, \dots, p_K, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_K) = \mathbb{E}_{\mathbf{x} \sim p}\left[\sum_{k=1}^{K} \Big(\boldsymbol{\tau}_k(\mathbf{x}) - \boldsymbol{\pi}_k\Big) \times \mathcal{C}_k(\mathbf{x})\right]$$

Synthetic Example



Model Selection

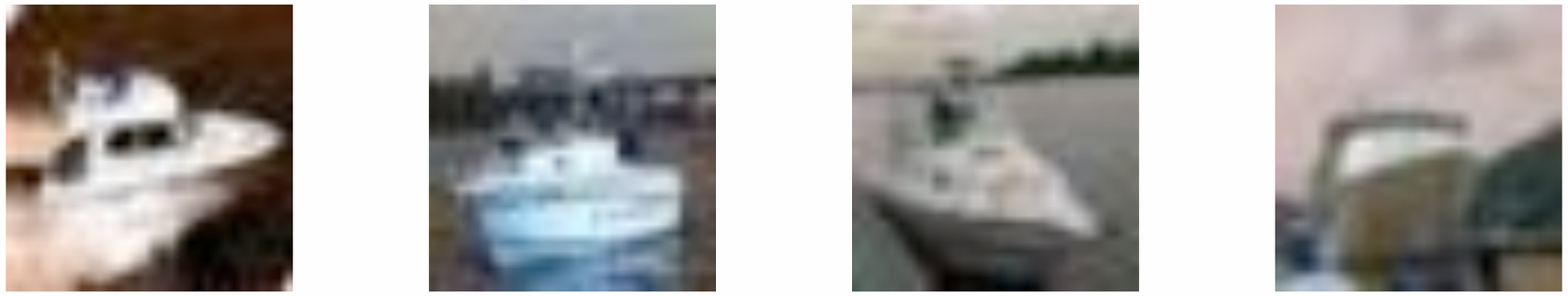
Wasserstein distances involved for GeWaC and DiWaC can be evaluated on held-out unlabelled data for model selection

Over-clustering experiments on real data are in progress

GeWaC on CIFAR-10

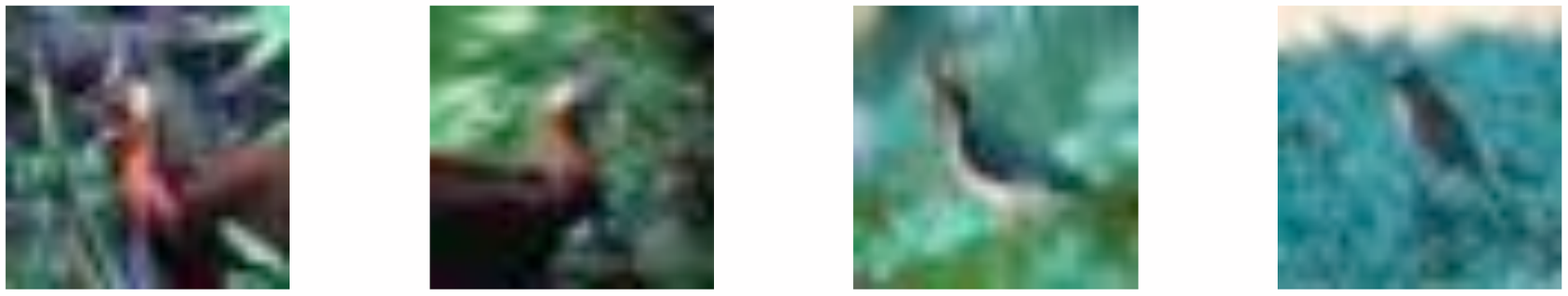
Generated Examples from a Cluster
(that *looks* like 'plane')

DiWaC on CIFAR-10



One *over*-cluster
(that *looks* like 'boat')

DiWaC on CIFAR-10



One *over*-cluster
(that *looks* like 'bird')

Experiments

Datasets	MNIST	Reuters	Reuters-10k	HHAR
DiWaC (ours)	**98.42**	**84.24**	**84.87**	**92.42**
GeWaC (ours with fixed proportions from AE+GMM)	**97.37**	**82.14**	**82.27**	**87.54**
ClusterGAN [Mukherjee et al., 2019]	90.97	–	–	–
VaDE [Jiang et al., 2016]	94.06	79.38	79.83	84.46
DEC [Xie et al., 2015]	84.30	75.63	72.17	79.82
AE + GMM (full covariance)	82.56	70.98	70.12	78.48
IMSAT [Hu et al., 2017]	98.40	–	71.00	–
GAR [Kilinc and Uysal, 2018]	98.32	–	–	–
DEPICT [Dizaji et al., 2017]	96.50	–	–	–
GMM (diagonal covariance)	53.73	55.81	54.72	60.34
k-Means	53.47	53.29	54.04	59.98

Experimental accuracy results (%, the higher, the better)
based on the Hungarian method.
(the last rows correspond to methods without neural networks)

Prediction with Uncertainty

Uncertainty Sources

Epistemology Choice for Supervised Learning

Experience for Image Classification

Uncertainty Sources

Extrapolation	Far from the training data *Cat feeding a car/plane classifier*
Aleatoric	Wrong or noisy information *Real-world thermometer*
Epistemic	Inherently uncertain prediction *Butterfly Effect*

Supervised Learning

$$\mathbf{y} \simeq \mathcal{F}(\mathbf{x})$$

$$\min_{\mathcal{F}} \mathcal{L}(\mathcal{F})$$

$$\mathcal{L}(\mathcal{F}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim \text{Data}}\Big(\ell(\mathbf{y}, \mathcal{F}(\mathbf{x}))\Big)$$

Supervised Learning

$$\mathbf{y} \simeq \mathcal{F}(\mathbf{x})$$

$$\min_{\mathcal{F}} \mathcal{L}(\mathcal{F})$$

$$\mathcal{L}(\mathcal{F}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim \text{Data}}\Big(\ell(\mathbf{y}, \mathcal{F}(\mathbf{x}))\Big)$$

Regression

$$\ell(\mathbf{y}, \mathbf{z}) = \|\mathbf{y} - \mathbf{z}\|_2^2$$

Supervised Learning

$$\mathbf{y} \simeq \mathcal{F}(\mathbf{x})$$

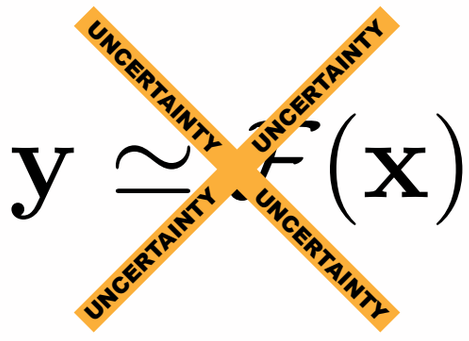
$$\min_{\mathcal{F}} \mathcal{L}(\mathcal{F})$$

$$\mathcal{L}(\mathcal{F}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim \text{Data}}\Big(\ell(\mathbf{y}, \mathcal{F}(\mathbf{x}))\Big)$$

Classification

$$\ell(\mathbf{y}, \mathbf{z}) = -\mathbf{y}^\top \log(\mathbf{z})$$

Uncertainty



$$\mathbf{y} \simeq \hat{F}(\mathbf{x})$$

$$\mathbf{y} \sim \mathcal{F}(\mathbf{x})$$

$$\min_{\mathcal{F}} \mathcal{L}(\mathcal{F})$$

Uncertainty

$$\mathbf{y} \sim \mathcal{F}(\mathbf{x})$$

$$\min_{\mathcal{F}} \mathcal{L}(\mathcal{F})$$

$$\ell(\mathbf{y}, \mathcal{F}_{\text{old}}(\mathbf{x})) = \mathbb{E}_{\mathbf{z}|\mathbf{x} \sim \delta_{\mathcal{F}_{\text{old}}(\mathbf{x})}} \big(\ell(\mathbf{y}, \mathbf{z})\big)$$

A Gentle Idea

$$\mathcal{L}(\mathcal{F}_{\text{old}}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\text{Data}}\Big(\ell(\mathbf{y}, \mathcal{F}_{\text{old}}(\mathbf{x}))\Big)$$

$$\ell(\mathbf{y}, \mathcal{F}_{\text{old}}(\mathbf{x})) = \mathbb{E}_{\mathbf{z}|\mathbf{x}\sim\delta_{\mathcal{F}_{\text{old}}(\mathbf{x})}}\Big(\ell(\mathbf{y}, \mathbf{z})\Big)$$

$$\mathcal{L}(\mathcal{F}_{\text{new}}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\text{Data}}\left(\mathbb{E}_{\mathbf{z}|\mathbf{x}\sim\mathcal{F}_{\text{new}}(\mathbf{x})}\Big(\ell(\mathbf{y}, \mathbf{z})\Big)\right)$$

Classification with Uncertainty

$$\mathcal{F}(\mathbf{x}) = \mathcal{U}(\mathcal{A}(\mathbf{x}), \mathcal{B}(\mathbf{x}))$$

$$\min_{\mathcal{F}} \frac{-1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{y}_i^\top \log(t_{i,j} \times \mathcal{A}(\mathbf{x}_i) + (1 - t_{i,j}) \times \mathcal{B}(\mathbf{x}_i))$$

$$t \sim \mathcal{U}(0, 1)$$

$$\min_{\mathcal{F}} \frac{-1}{N} \sum_{i=1}^{N} \mathbf{y}_i^\top \log(\mathcal{F}(\mathbf{x}_i))$$

Dogs *vs.* Wolves ... and Cats

Top-8 of most uncertain test images of our "Dogs *vs.* Wolves ... and Cats" dataset

Unsupervised Features Importance

$$\mathbf{x} \sim p$$
$$\mathbf{y} \sim q_{t,\mathcal{F}} : \mathbf{x}' \sim p \quad \text{and } \mathbf{y} = \mathbf{x}' + t \times \mathcal{F}(\mathbf{x}')$$

$$\max_{\substack{d_\phi \\ \mathcal{F} \text{ s.t. } \|\mathcal{F}\|_2=1}} \left| \nabla_{t=0} \left(W_{d_\phi}(p, q_{t,\mathcal{F}}) \right) \right|$$

$$\begin{aligned} d_\phi : \quad \mathbb{R}^D \times \mathbb{R}^D &\to \mathbb{R}_+ \\ (\mathbf{x}, \mathbf{y}) &\mapsto \|\phi(\mathbf{y}) - \phi(\mathbf{x})\|_2 \end{aligned}$$

Unsupervised Features Importance

Preliminary Experiment

Conclusion

Input Cardinality

Clustering

Output Interpretation

Uncertainty

Input Dimensionality

Features

Representation is key

Clustering is ill-posed

Impossibility theorem for Clustering
Kleinberg, 2002

No clustering algorithm can simultaneously verify these 3 properties

Scale Invariance	*e.g.* neighbourhood threshold fails
Cluster Shapes Invariance	*e.g.* k-Means fails on the Moons data
Metric Invariance	*e.g.* broken pairwise relationships

Others (1/2)

Cerbero

Applied AI for real-time credit card fraud detection
with pragmatic and cost-oriented optimization
Extension of Prediction with Uncertainty *chapter with colleagues Roland Thiollière, Romain Nio, Nils Grunwald, Jérémie Thomas, Julien Gaunon, and Dr. Stéphane Raux (random order)*

NMJ

Reconciliation between grammar from linguistics and neural networks via auto-encoders and modern embedding techniques
Internship supervision of Maxime Haddouche

i2nn

Invitation to Neural Networks, talk given several times to convince people working both in Statistics and Programming to use Deep Learning
Material for the State of the Art *manuscript*

Artificial Intelligence Watch

Several talks given at Oscaro.com about Artificial Intelligence with academic and corporate points of view
Material for the State of the Art *manuscript*

Others (2/2)

Wasserstein Co-Clustering

Computational Biology for studying the Huntington disease using co-clustering on RNA data
Extension of the Wasserstein Clustering *chapter with Thi Thanh Yen Nguyen, Dr. Olivier Bouaziz and Pr. Antoine Chambaz*

GaDeMI

New kind of auto-encoders built in successive layers
to extract a representation whose coordinates are quasi-gaussian and decorrelated (and approximately independent)
Extension of appendix for Images with Dr. Joan Alexis Glaunès

StaReLefOU

State Representation Learning for Robotics Exploration
Extension of the Prediction with Uncertainty *chapter with Astrid Merckling*

Thank you