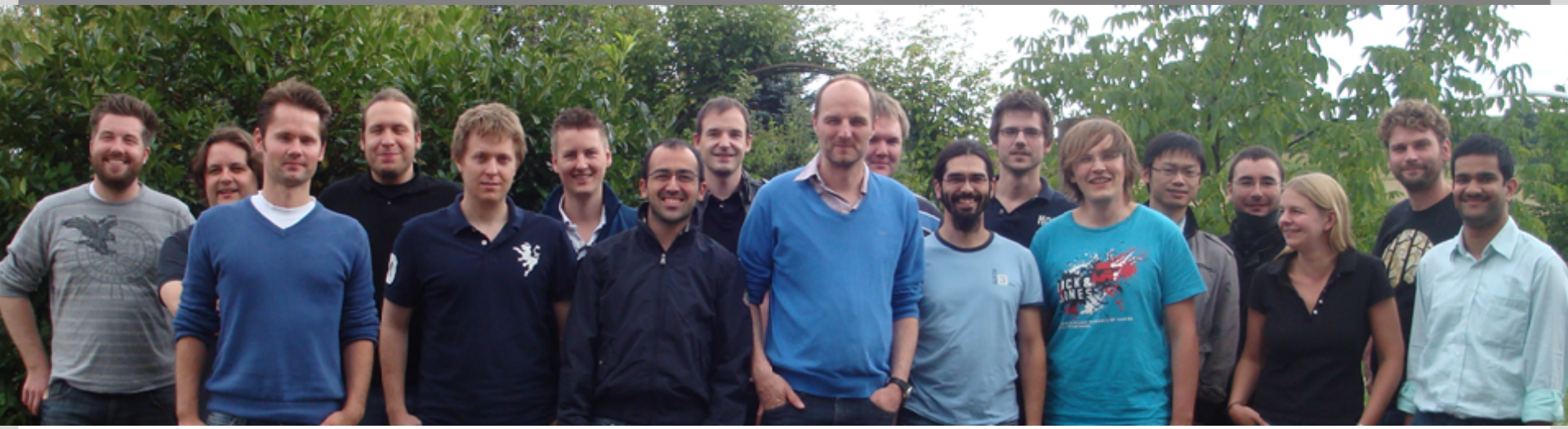
Karlsruhe Institute of Technology

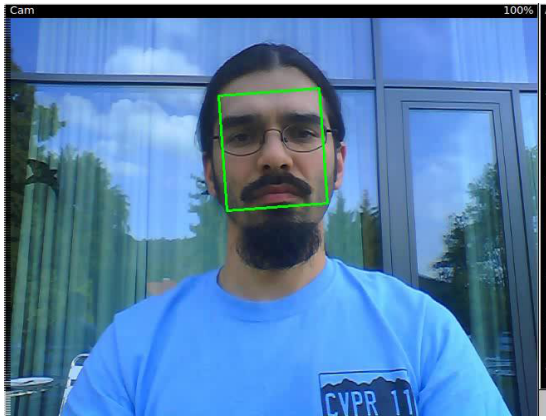
The Wasserstein Distance as a Loss Function in Deep Learning

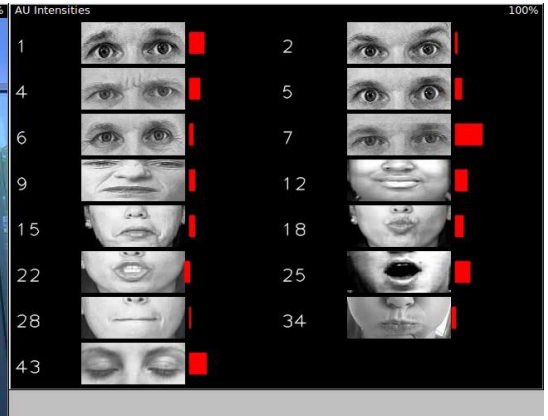
Dr-Ing. Manuel Martinez

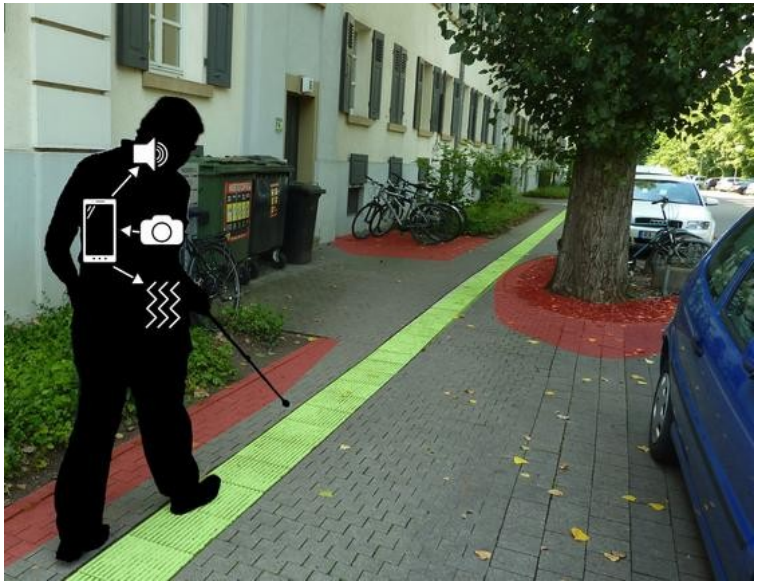
INSTITUTE FOR ANTHROPOMATICS, CV-HCI



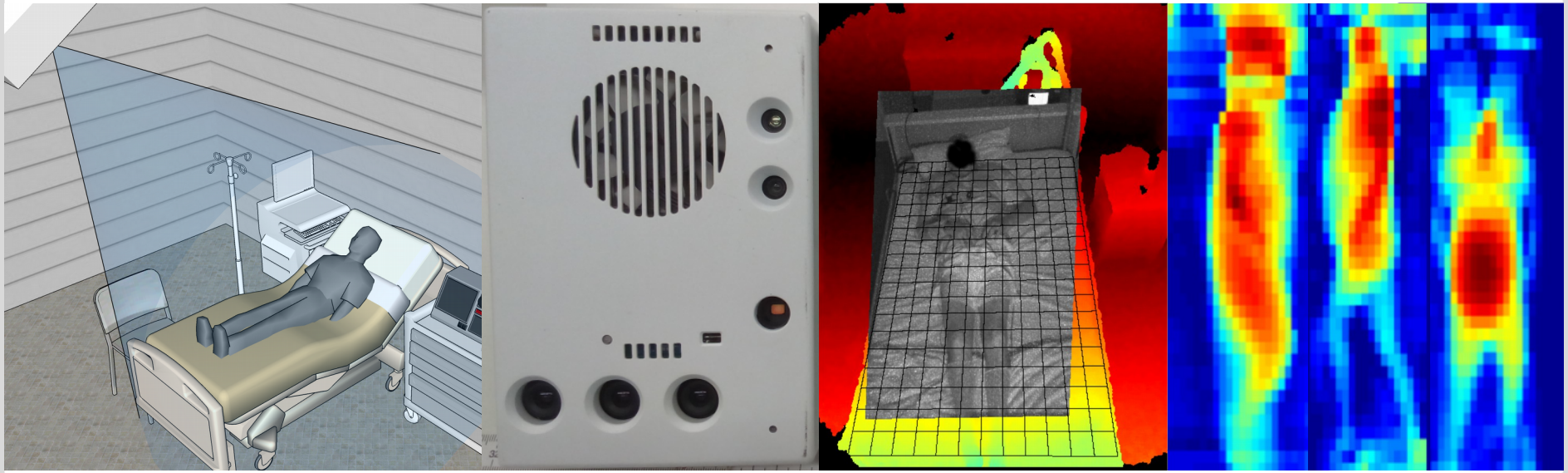
Computer Vision for Human Computer Interaction
Prof. Rainer Stiefelhagen

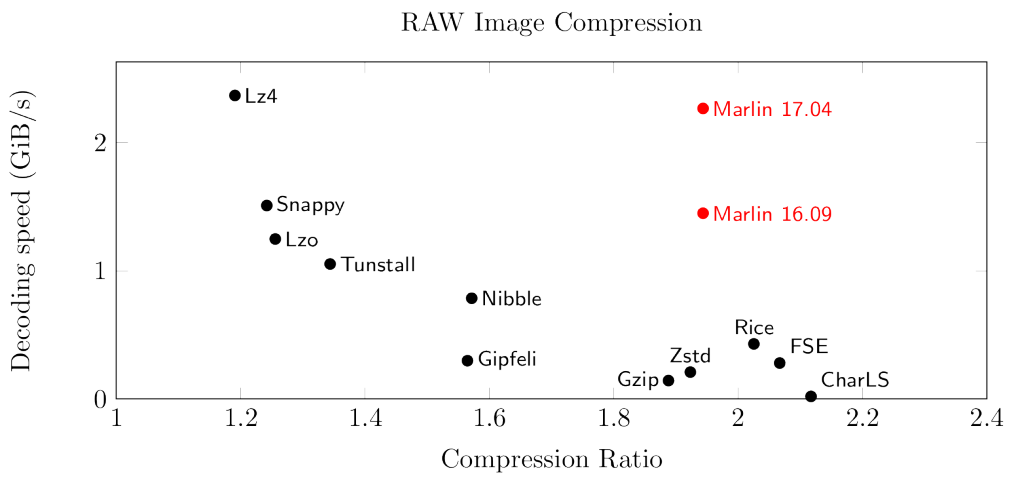


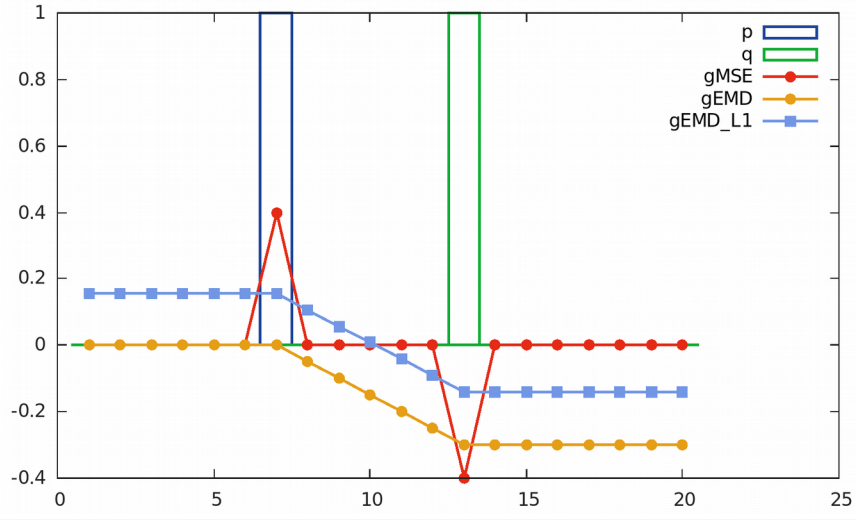




What do I do?




RAW Image Compression
Decoding speed (GiB/s)
Compression Ratio
Lz4
Snappy
Lzo
Tunstall
Nibble
Gipfeli
Gzip
Zstd
Rice
FSE
CharLS
Marlin 17.04
Marlin 16.09
0
1
2
1
1.2
1.4
1.6
1.8
2
2.2
2.4


p
q
gMSE
gEMD
gEMD_L1
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
5
10
15
20
25

Please, fasten your seatbelts in preparation for:

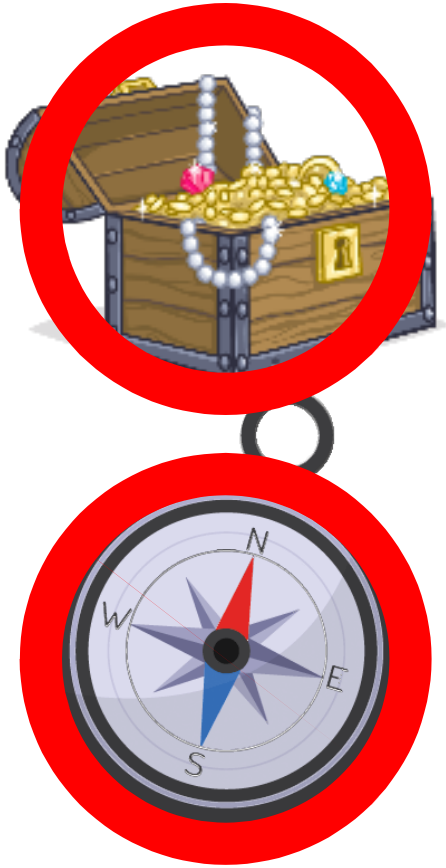
The Wonderful World of Optimal Transport Theory

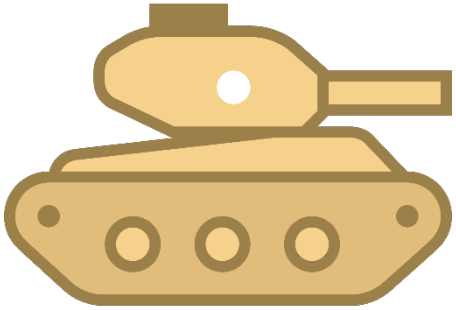
Loss Functions a.k.a. criterion

Analogy time!

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

Analogy time!



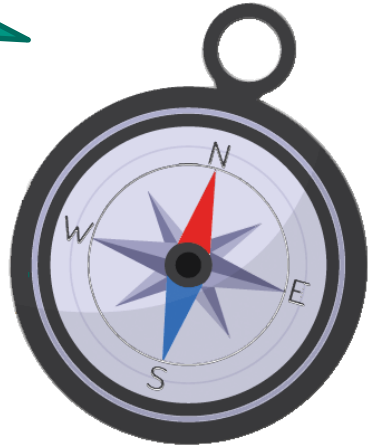


Analogy time!

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

$$pos_{n+1} = pos_n - \gamma \nabla_{pos_n} L(pos_n, target)$$

L: Loss Function
(a.k.a) Criterion



Closer look:

- Loss Functions for Classification:
  - Logistic Regression, Logit, CrossEntropy, Negative Log Likelihood:

$$H(p, q) = -\sum_{x} p(x) \log q(x).$$

  - others (that basically nobody uses)

- Loss Functions for Regression:
  - L1 distance
  - MSE distance
  - Kullback-Leibler divergence:

$$D_{\mathrm{KL}}(P \| Q) = -\sum_{i} P(i) \log \frac{Q(i)}{P(i)},$$

Wasserstein Distance

(images from Marco Cuturi's book)

Origins: Monge Problem (1781)

MÉMOIRE
SUR LA
THÉORIE DES DÉBLAIS
ET DES REMBLAIS.

Par M. MONGE.

LORSQU'ON doit tranſporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit tranſporter, & le nom de *Remblai* à l'eſpace qu'elles doivent occuper après le tranſport.

Kantorovich Problem

Kantorovich

Tolstoi

1930

Hitchcock

1939

THE DISTRIBUTION OF A PRODUCT FROM SEVERAL SOURCES TO NUMEROUS LOCALITIES

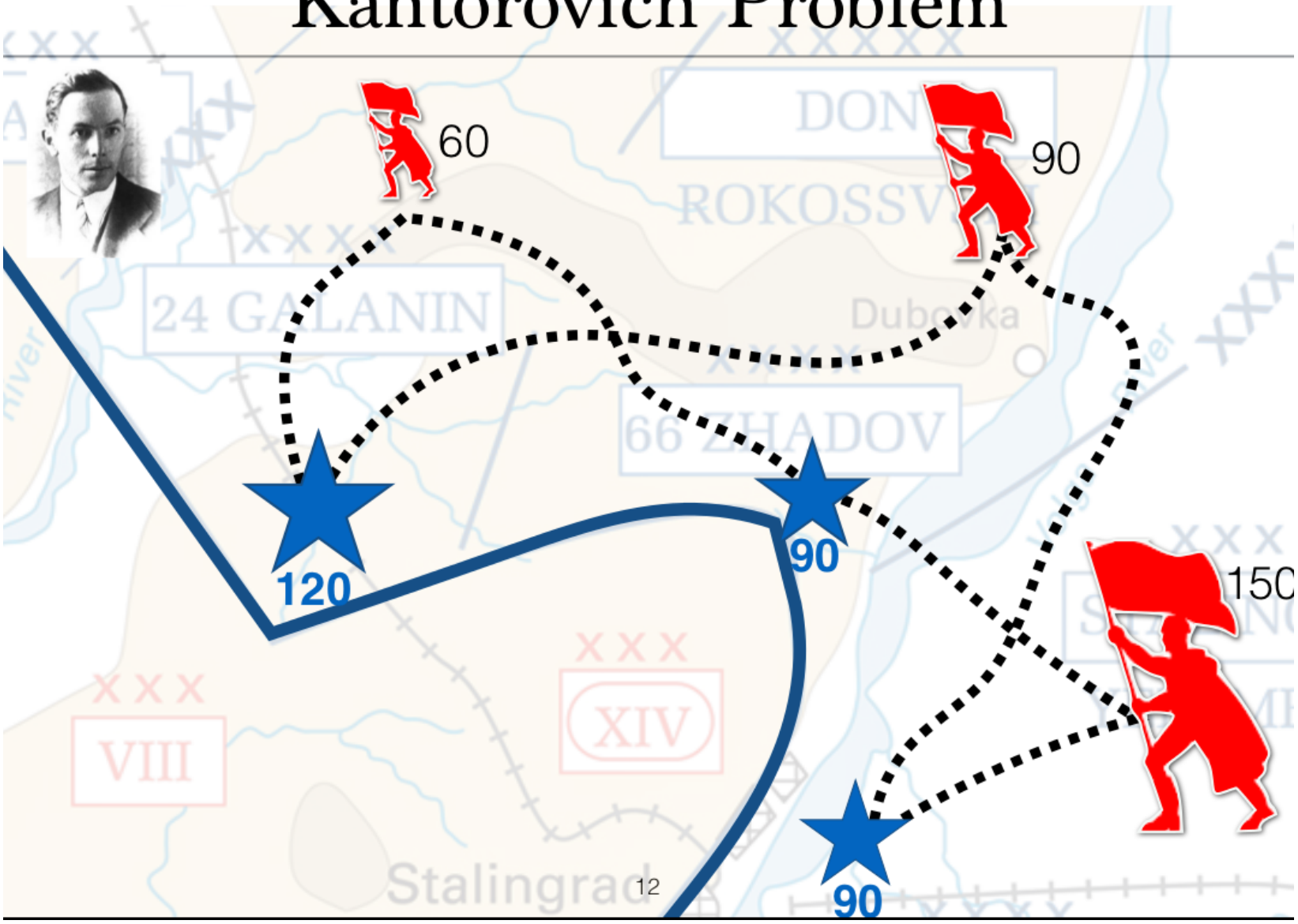
By Frank L. Hitchcock

1. Statement of the problem. When several factories supply a product to a number of cities we desire the least costly manner of distribution. Due to freight rates and other matters the cost of a ton of product to a particular city will vary according to which factory supplies it, and will also vary from city to city.

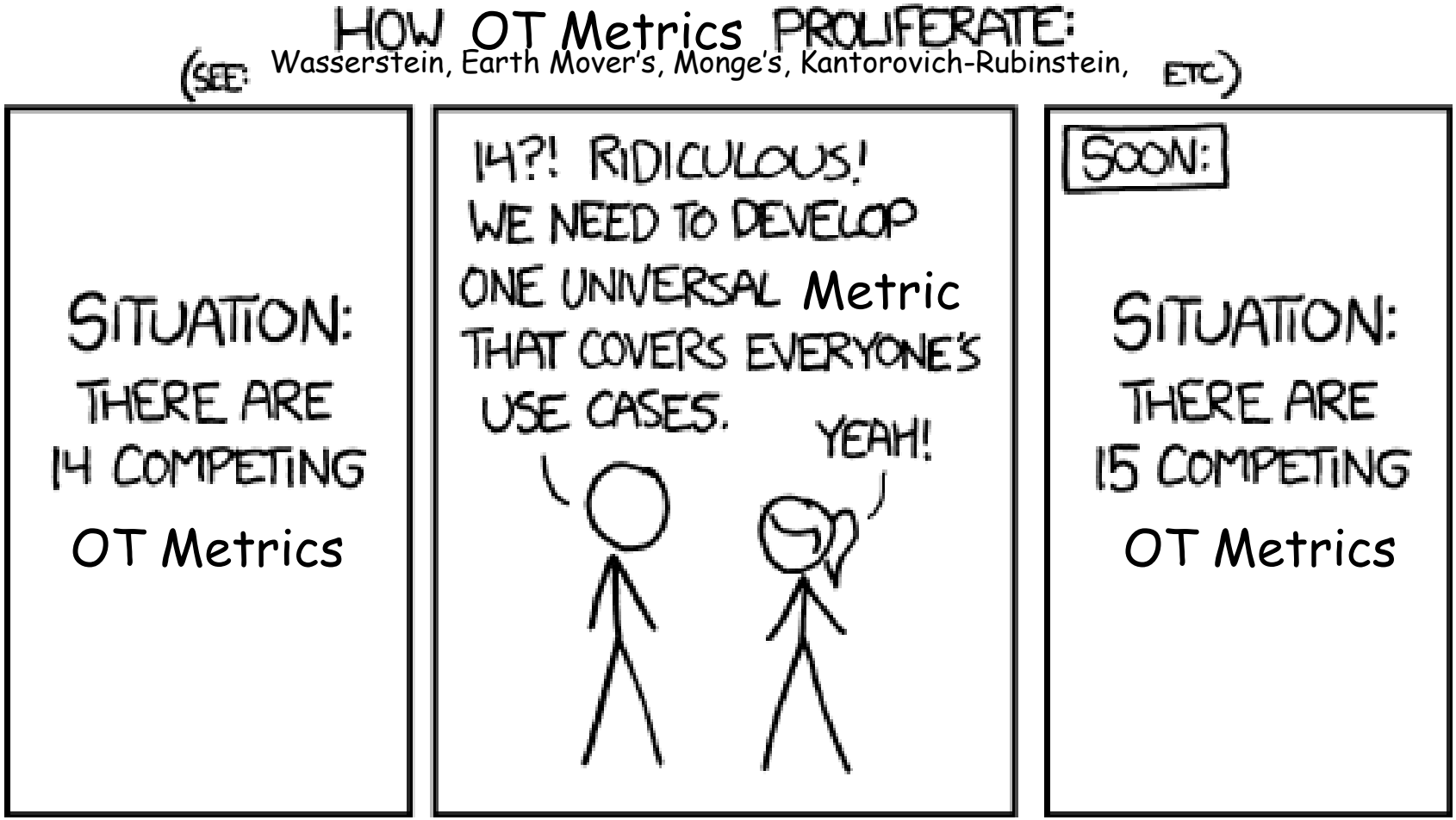
1941

11

Kantorovich Problem



Optimal Transport Theory.
But… there is a naming conflict.



Wikipedia:

In mathematics, the **Wasserstein** or **Kantorovich-Rubinstein metric** or **distance** is a distance function defined between probability distributions on a given metric space M. It is called the **Kantorovich-Monge-Rubinstein metric** or **distance**.

Intuitively, if each distribution is viewed as a unit amount of "dirt" piled on M, the metric is the minimum "cost" of turning one pile into the other, which is assumed to be the amount of dirt that needs to be moved times the distance it has to be moved. Because of this analogy, the metric is known in computer science as the earth mover's distance.

The name "Wasserstein distance" was coined by R. L. Dobrushin in 1970, after the Russian mathematician Leonid Vaserštein who introduced the concept in 1969. Most English-language publications use the German spelling "Wasserstein" (attributed to the name "Vaserstein" being of German origin).

Formulations

$$W_p(\mu,\nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{M\times M} d(x,y)^p \, \mathrm{d}\gamma(x,y) \right)^{1/p}$$

where $\Gamma(\mu,\nu)$ denotes the collection of all measures on $M \times M$ with marginals μ and ν on the first and second factors respectively. (The set $\Gamma(\mu,\nu)$ is also called the set of all **couplings** of μ and ν.)

The above distance is usually denoted $W_p(\mu,\nu)$ (typically among authors who prefer the "Wasserstein" spelling) or $\ell_p(\mu,\nu)$ (typically among authors who prefer the "Vaserstein" spelling).

2.2. **Optimal Transportation.** Given a $d \times d$ cost matrix M, the cost of mapping r to c using a transportation matrix (or joint probability) P can be quantified as $\langle P, M \rangle$. The following problem:

$$d_M(r,c) \stackrel{\text{def}}{=} \min_{P \in U(r,c)} \langle P, M \rangle.$$

Sinkhorn Distances: Marco Cuturi
Lightspeed Computation of OT (NIPS 2013)

- Solving OT is generally costly:

$$d_M(r,c) \stackrel{\text{def}}{=} \min_{P \in U(r,c)} \langle P, M \rangle.$$

- By adding an entropic regularization term, it can be made convex:

$$d_M^\lambda(r,c) \stackrel{\text{def}}{=} \langle P^\lambda, M \rangle, \quad \text{where } P^\lambda = \operatorname*{argmin}_{P \in U(r,c)} \langle P, M \rangle - \frac{1}{\lambda} h(P)$$

Sinkhorn Distances: Marco Cuturi
Lightspeed Computation of OT (NIPS 2013)

Algorithm 1 Computation of $d_M^\lambda(r, c)$ using Sinkhorn-Knopp's fixed point iteration

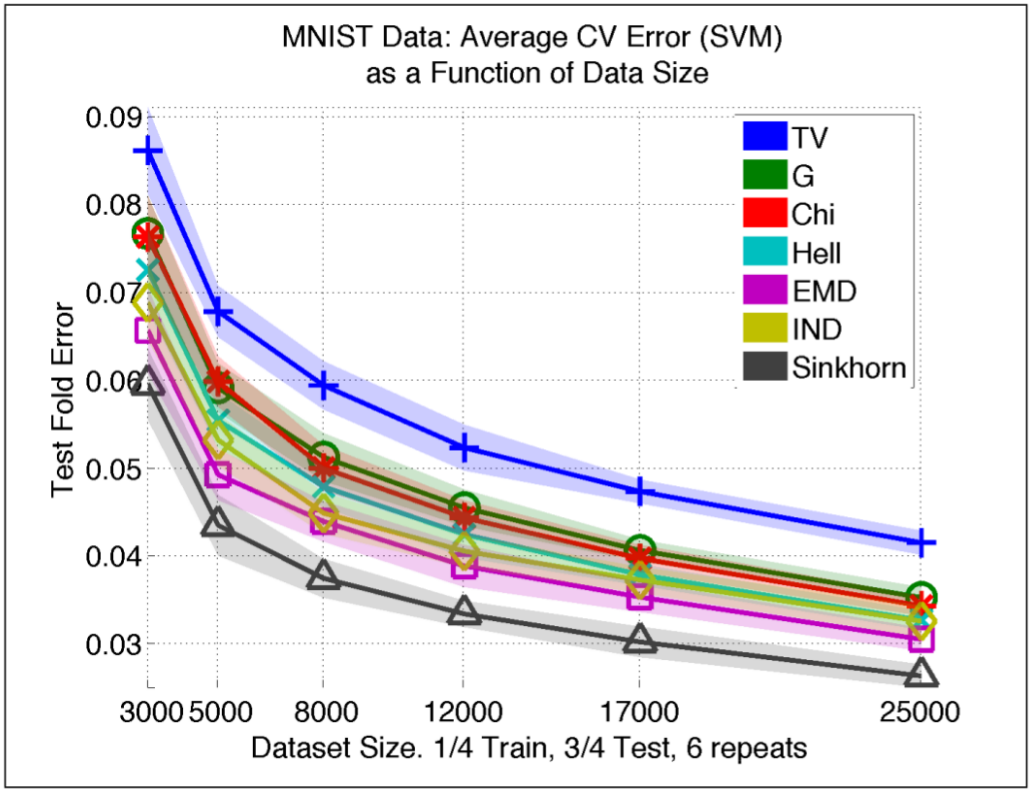
Input M, λ, r, c.

```
I=(r>0); r=r(I); M=M(I,:); K=exp(-λ*M)
Set x=ones(length(r),size(c,2))/length(r);
while x changes do
  x=diag(1./r)*K*(c.*(1./(K'*(1./x))))
end while
u=1./x; v=c.*(1./(K'*u))
```

d_M^λ(r,c)=sum(u.*((K.*M)*v))

Marco Cuturi provides a GPU implementation

Sinkhorn Distances: Marco Cuturi Lightspeed Computation of OT (NIPS 2013)



SVM classification with kernel: exp(-d/t).

Note how Sinkhorn is better than EMD.

Learning with a Wasserstein Loss. Frogner et al. NIPS 2015

- To use Sinkhorn Distance as a Loss, we need its gradient.

Algorithm 1 Gradient of the Wasserstein loss

Given $h(x)$, y, λ, $\mathbf{K}$.
$u \leftarrow \mathbf{1}$
while u has not converged **do**
$$u \leftarrow h(x) \oslash \left(\mathbf{K}\left(y \oslash \mathbf{K}^\top u\right)\right)$$
end while
$$\partial W_p^p / \partial h(x) \leftarrow \frac{\log u}{\lambda} - \frac{\log u^\top \mathbf{1}}{\lambda K}\mathbf{1}$$

Learning with a Wasserstein Loss. Frogner et al. NIPS 2015

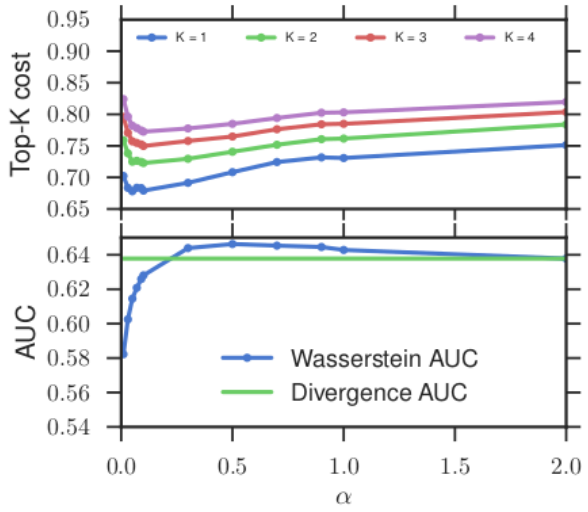
(a) **Flickr user tags**: street, parade, dragon; **our proposals**: people, protest, parade; **baseline proposals**: music, car, band.

(b) **Flickr user tags**: water, boat, reflection, sunshine; **our proposals**: water, river, lake, summer; **baseline proposals**: river, water, club, nature.

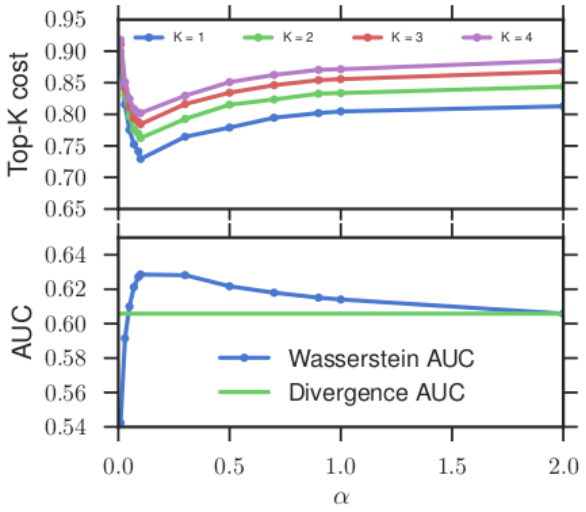
Figure 7: Examples of images in the Flickr dataset. We show the groundtruth tags and as well as tags proposed by our algorithm and the baseline.

Learning with a Wasserstein Loss. Frogner et al. NIPS 2015

1. Extract deep features from the images
2. Learn a multiclass classification with the tags using a mix of KL and Wasserstein Losses
3. Minimization is not actually mentioned in the paper, but the method of alternating projections is mentioned.



(a) Original Flickr tags dataset.



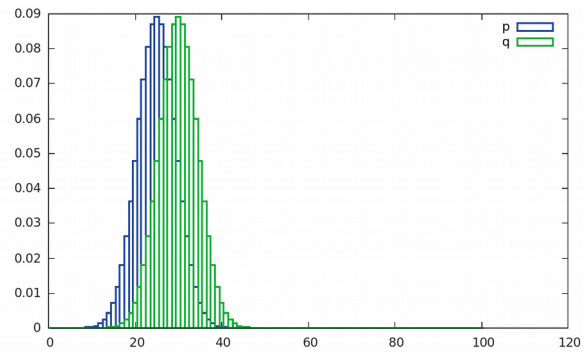
(b) Reduced-redundancy Flickr tags dataset.

Karlsruhe Institute of Technology

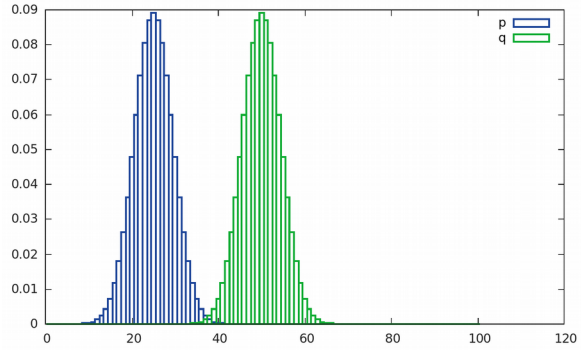
special case for 1D distributions

Earth Mover's Distance in 1D distributions

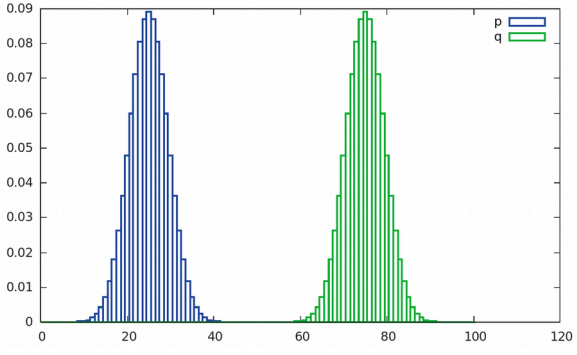
- Is the de-facto L1 distance between histograms.



EMD: 5



EMD: 25



EMD: 50

- Closed form solution for EMD:

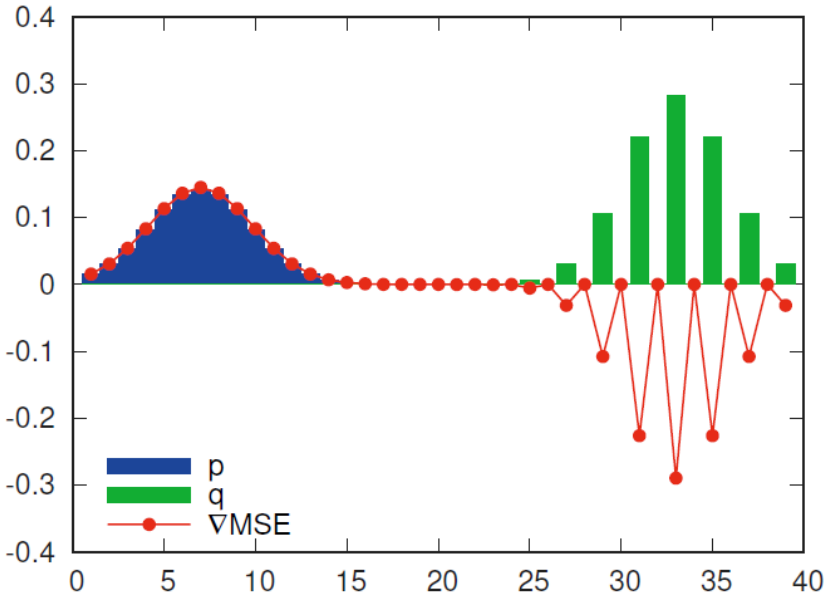
$$\sum_{i=1}^{N} |\varphi_i|, \text{ where } \varphi_i = \sum_{j=1}^{i} \left(\frac{a_j}{\|\mathbf{a}\|_1} - \frac{b_j}{\|\mathbf{b}\|_1} \right)$$

- Closed form for ∇EMD:

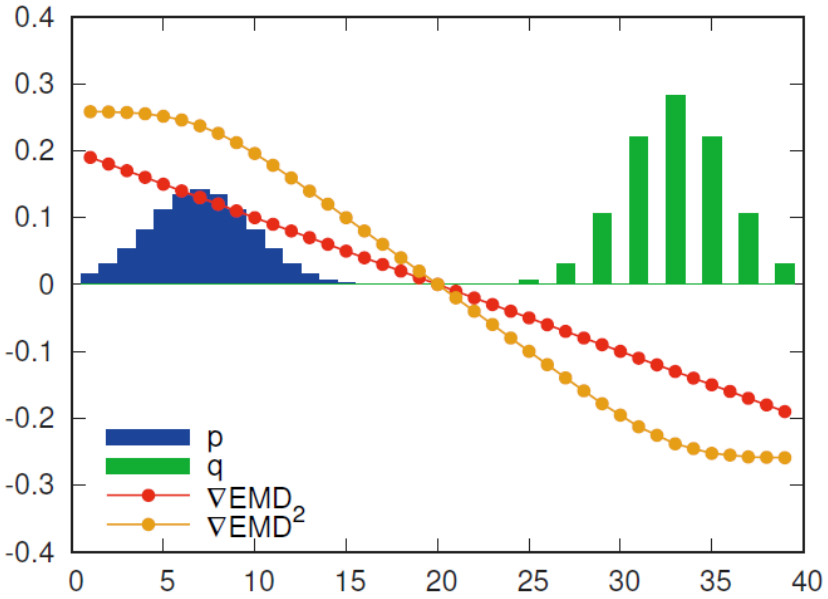
$$\sum_{i=1}^{N} \text{sgn}(\varphi_i) \sum_{j=1}^{i} (N\delta_{jk} - 1)$$

"A Closed-form Gradient for the 1D Earth Mover's Distance for Spectral Deep Learning on Biological Data", Martinez, ICMLw 2016

How does the Gradient Look Like?

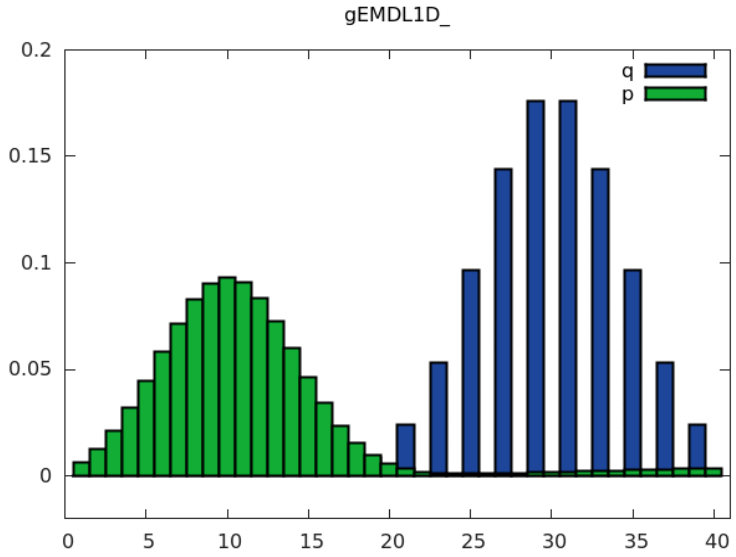


(a) Mean Squared Error

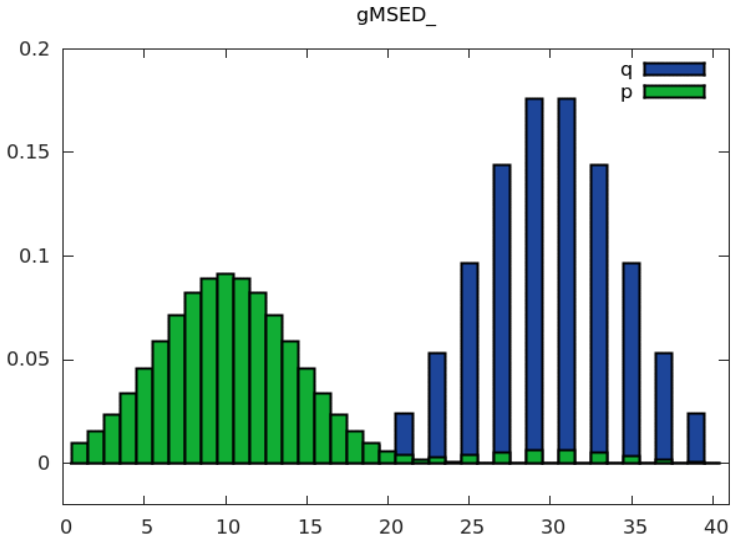


(b) Earth Mover's Distance

MSE vs Earth Mover's Distance: Smoothness

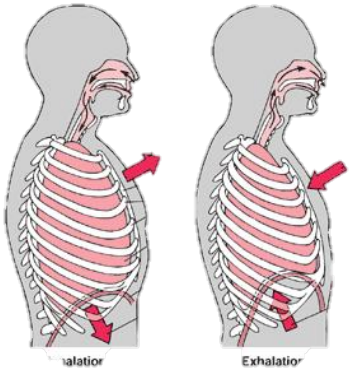


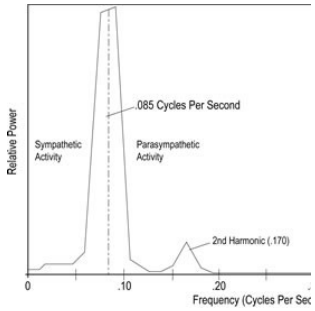
EMD

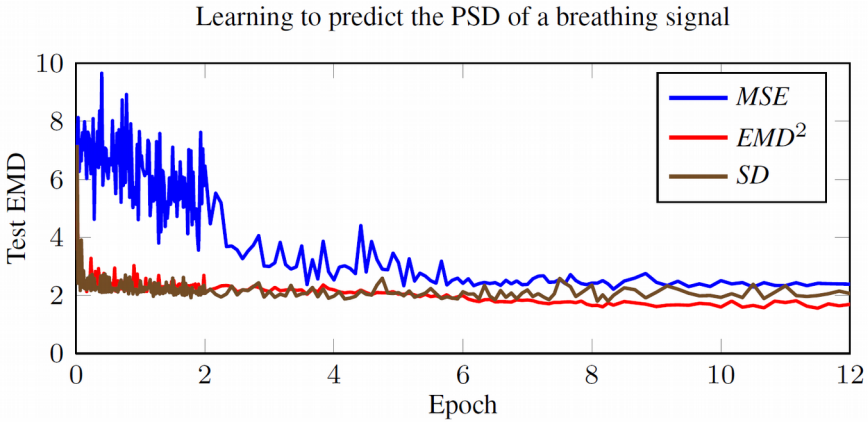


MSE

Example: Breathing Rate

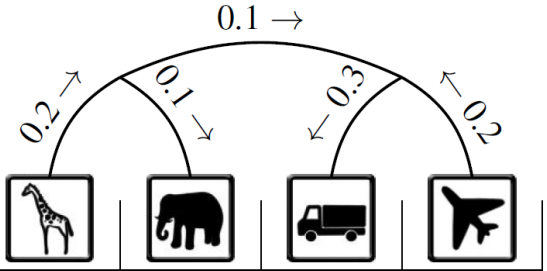






"Relaxed Earth Mover's Distances for Chain-and Tree-connected Spaces and their use as a Loss Function in Deep Learning", Martinez, arxiv, 2016

Hierarchical Earth Mover's Distance



f ()	0.2	0.4	0.2	0.2
y 	0.0	0.5	0.5	0.0
∇EMD	1.5	−0.5	−1.5	0.5
∇EMD^2	0.5	−0.1	−0.7	0.3

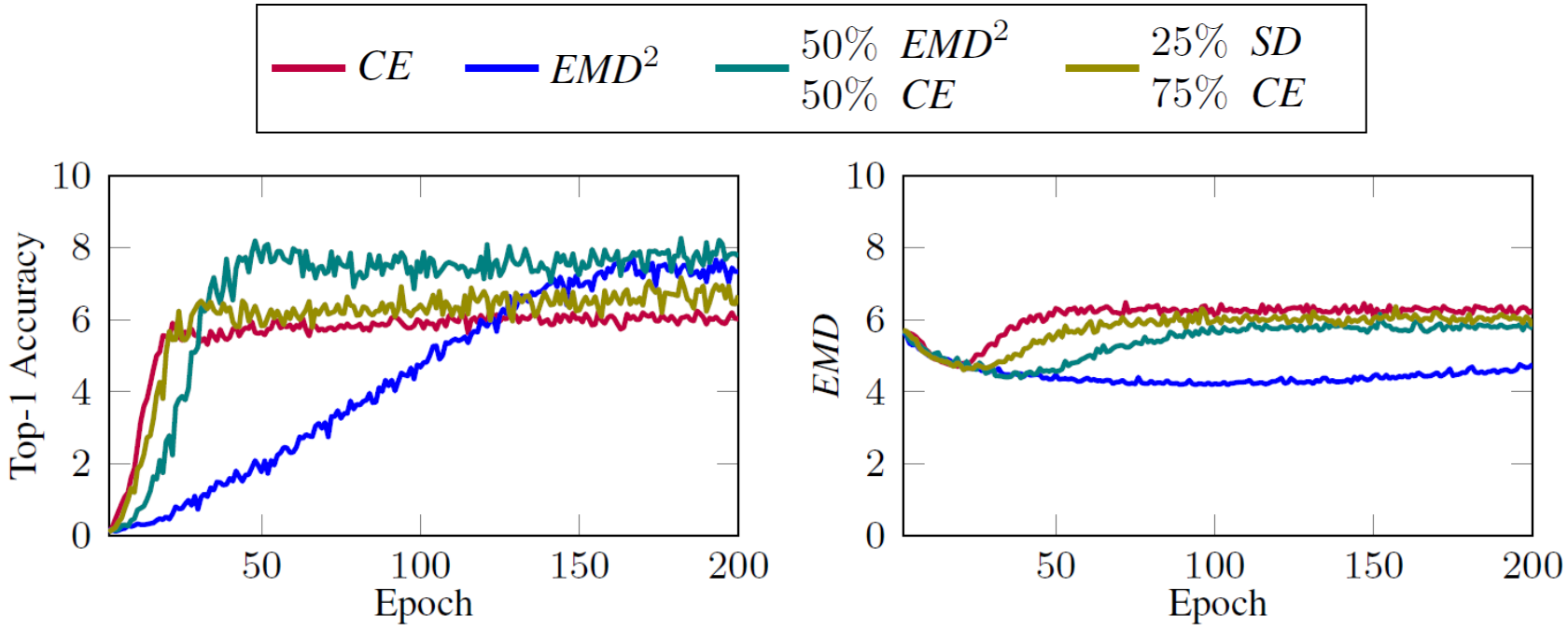
$$HEMD^{\rho}(\mathbf{p},\mathbf{q}) = \sum_{i \in G} M_{i,\mathbf{p}(i)} \cdot |\tilde{\varphi}_i|^{\rho},$$

$$\tilde{\varphi}_i = \sum_{j \in \mathbf{l}(i)} (p_j - q_j) \ ,$$

$$\nabla HEMD^{\rho} \simeq \rho \sum_{i \in G} \tilde{M}_i \cdot \tilde{\varphi}_i \cdot |\tilde{\varphi}_i|^{\rho-2} \sum_{j=1}^{i} (\delta_{jk} - 1/N)$$

"Relaxed Earth Mover's Distances for Chain-and Tree-connected Spaces and their use as a Loss Function in Deep Learning", Martinez, arxiv, 2016

Example: ImageNet



(b) Top-1 Accuracy on ImageNet@50K images

(c) Holistic Error on ImageNet@50K images

"Relaxed Earth Mover's Distances for Chain-and Tree-connected Spaces and their use as a Loss Function in Deep Learning", Martinez, arxiv, 2016

Applications

From Word Embeddings to Document Distances. Kusner et al. ICML 2015

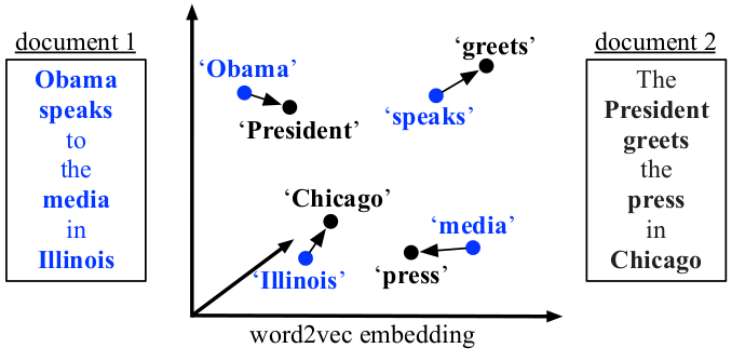


Figure 1. An illustration of the *word mover's distance*. All non-stop words (***bold***) of both documents are embedded into a *word2vec* space. The distance between the two documents is the minimum cumulative distance that all words in document 1 need to travel to exactly match document 2. (Best viewed in color.)

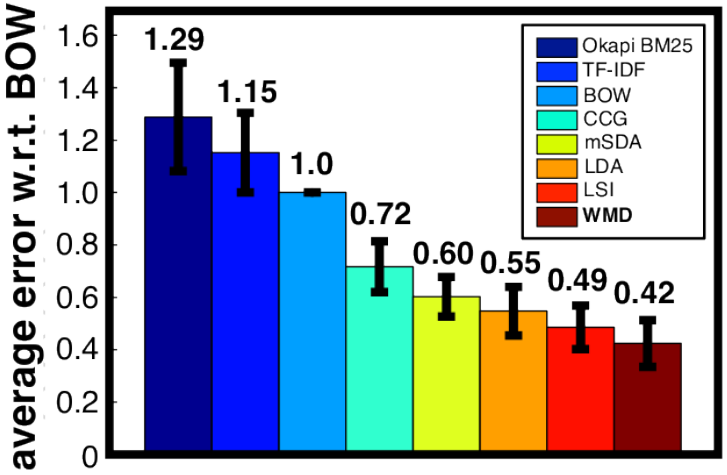


Figure 4. The kNN test errors of various document metrics averaged over all eight datasets, relative to kNN with BOW.

Joint distribution OT for domain adaptation (NIPS 2017)

The main idea of this work is is to handle a change in both marginal and conditional distributions. As such, we are looking for a transformation $\mathcal{T}$ that will align directly the joint distributions $\mathcal{P}_s$ and $\mathcal{P}_t$. Following the Kantovorich formulation of (2), $\mathcal{T}$ will be implicitly expressed through a coupling between both joint distributions as:

$$\boldsymbol{\gamma}_0 = \underset{\boldsymbol{\gamma} \in \Pi(\mathcal{P}_s, \mathcal{P}_t)}{\operatorname{argmin}} \int_{(\Omega \times \mathcal{C})^2} \mathcal{D}(\mathbf{x}_1, y_1; \mathbf{x}_2, y_2) d\boldsymbol{\gamma}(\mathbf{x}_1, y_1; \mathbf{x}_2, y_2), \tag{3}$$

where $\mathcal{D}(\mathbf{x}_1, y_1; \mathbf{x}_2, y_2) = \alpha \mathrm{d}(\mathbf{x}_1, \mathbf{x}_2) + \mathcal{L}(y_1, y_2)$ is a joint cost measure combining both the distances between the samples and a loss function $\mathcal{L}$ measuring the discrepancy between y_1 and y_2.

...

labels to the aligned target instances in the transport plan. For this purpose, we propose to solve the following problem for JDOT:

$$\min_{f, \boldsymbol{\gamma} \in \Delta} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) \boldsymbol{\gamma}_{ij} \quad \equiv \quad \min_f W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t^f) \tag{5}$$

Joint distribution OT for domain adaptation (NIPS 2017)

Table 1: Accuracy on the Caltech-Office Dataset. Best value in bold.

Domains	**Base**	**SurK**	**SA**	**ARTL**	**OT-IT**	**OT-MM**	**JDOT**
caltech→amazon	92.07	91.65	90.50	92.17	89.98	**92.59**	91.54
caltech→webcam	76.27	77.97	81.02	80.00	80.34	78.98	**88.81**
caltech→dslr	84.08	82.80	85.99	88.54	78.34	76.43	**89.81**
amazon→caltech	84.77	84.95	85.13	85.04	85.93	**87.36**	85.22
amazon→webcam	79.32	81.36	**85.42**	79.32	74.24	85.08	84.75
amazon→dslr	86.62	87.26	**89.17**	85.99	77.71	79.62	87.90
webcam→caltech	71.77	71.86	75.78	72.75	**84.06**	82.99	82.64
webcam→amazon	79.44	78.18	81.42	79.85	89.56	90.50	**90.71**
webcam→dslr	96.18	95.54	94.90	**100.00**	99.36	99.36	98.09
dslr→caltech	77.03	76.94	81.75	78.45	**85.57**	83.35	84.33
dslr→amazon	83.19	82.15	83.19	83.82	**90.50**	**90.50**	88.10
dslr→webcam	96.27	92.88	88.47	**98.98**	96.61	96.61	96.61
Mean	83.92	83.63	85.23	85.41	86.02	86.95	**89.04**
Mean rank	5.33	5.58	4.00	3.75	3.50	2.83	2.50
p-value	< 0.01	< 0.01	0.01	0.04	0.25	0.86	–

OT for Deep Joint Transfer Learning.
Lu et al. Sep. 2017

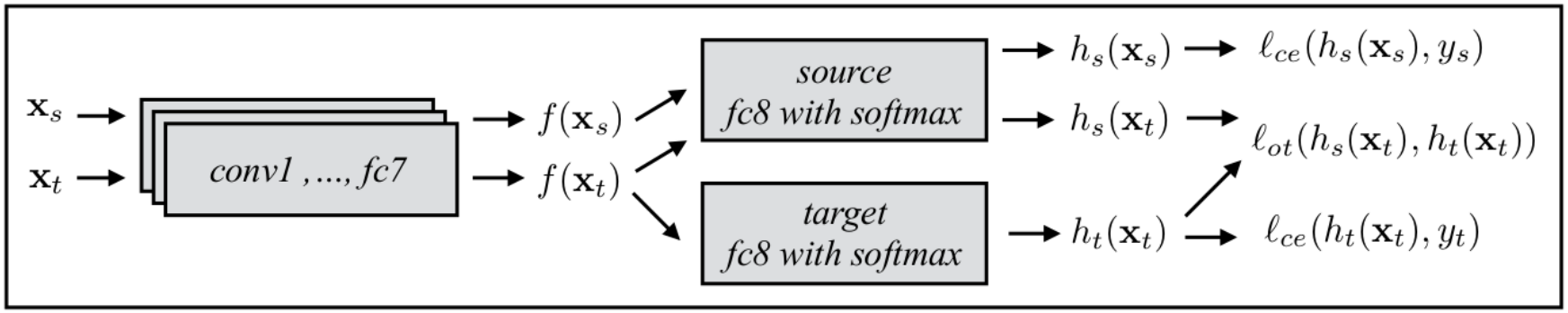


Figure 1: The structure and data flow of a Joint Transfer Learning Network based on Alexnet

OT for Deep Joint Transfer Learning (Sep. 2017)

Table 2: Experimental Results on the ITL Datasets (results are multi-class classification accuracy)

Methods	Boeing	Airbus
Finetuning on target	0.4796	0.4965
Consecutive finetuning on source+target	0.5286	0.545
Joint finetuning on source+target	0.5395	0.5497
JTLN (fc7MKMMD)	0.5422	**0.5982**
JTLN (fc7OT)	**0.5436**	0.5704

Learning Wasserstein Embeddings. Courty et al. (ICLR 2018)

Idea: fast computation of Wasserstein on images.

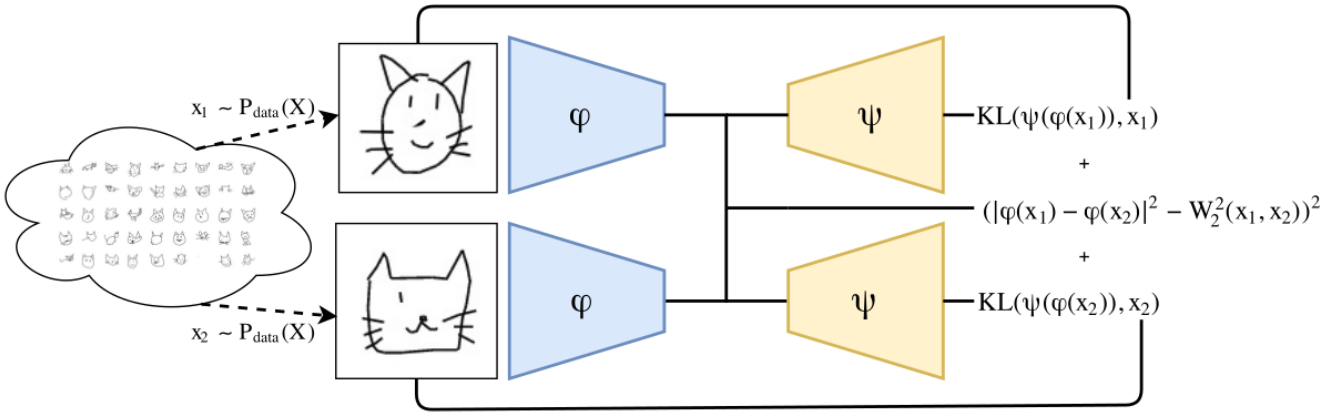


Figure 1: Architecture of the Deep Wasserstein Embedding: two samples are drawn from the data distribution and set as input of the same network (ϕ) that computes the embedding. The embedding is learnt such that the squared Euclidean distance in the embedding mimics the Wasserstein distance. The embedded representation of the data is then decoded with a different network (ψ), trained with a Kullback-Leibler divergence loss.

Learning Wasserstein Embeddings. Courty et al. (ICLR 2018)

Example use: linear interpolation of embeddings.

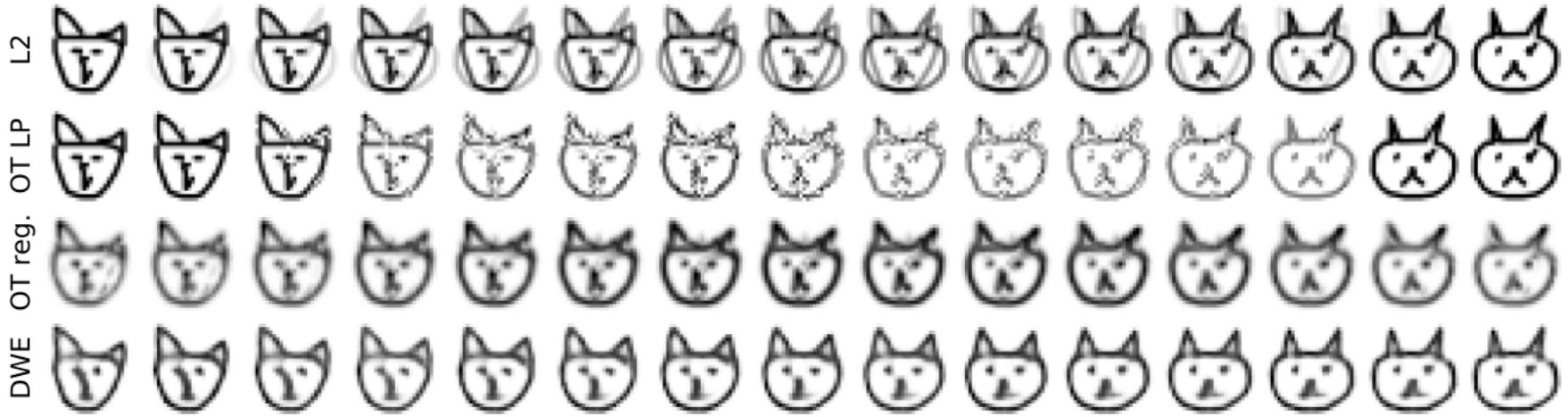
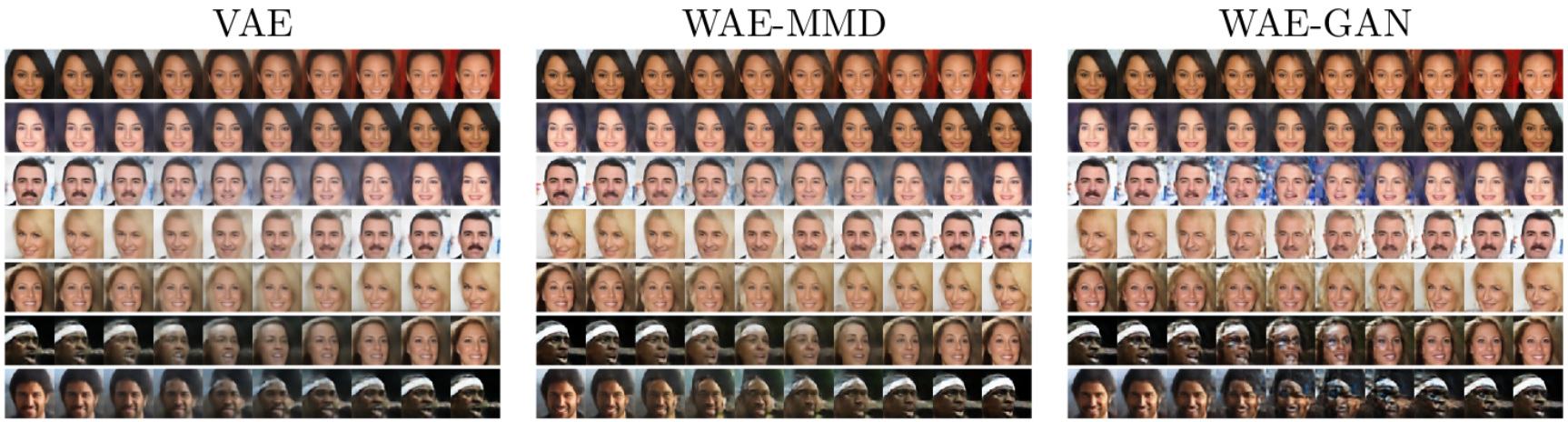


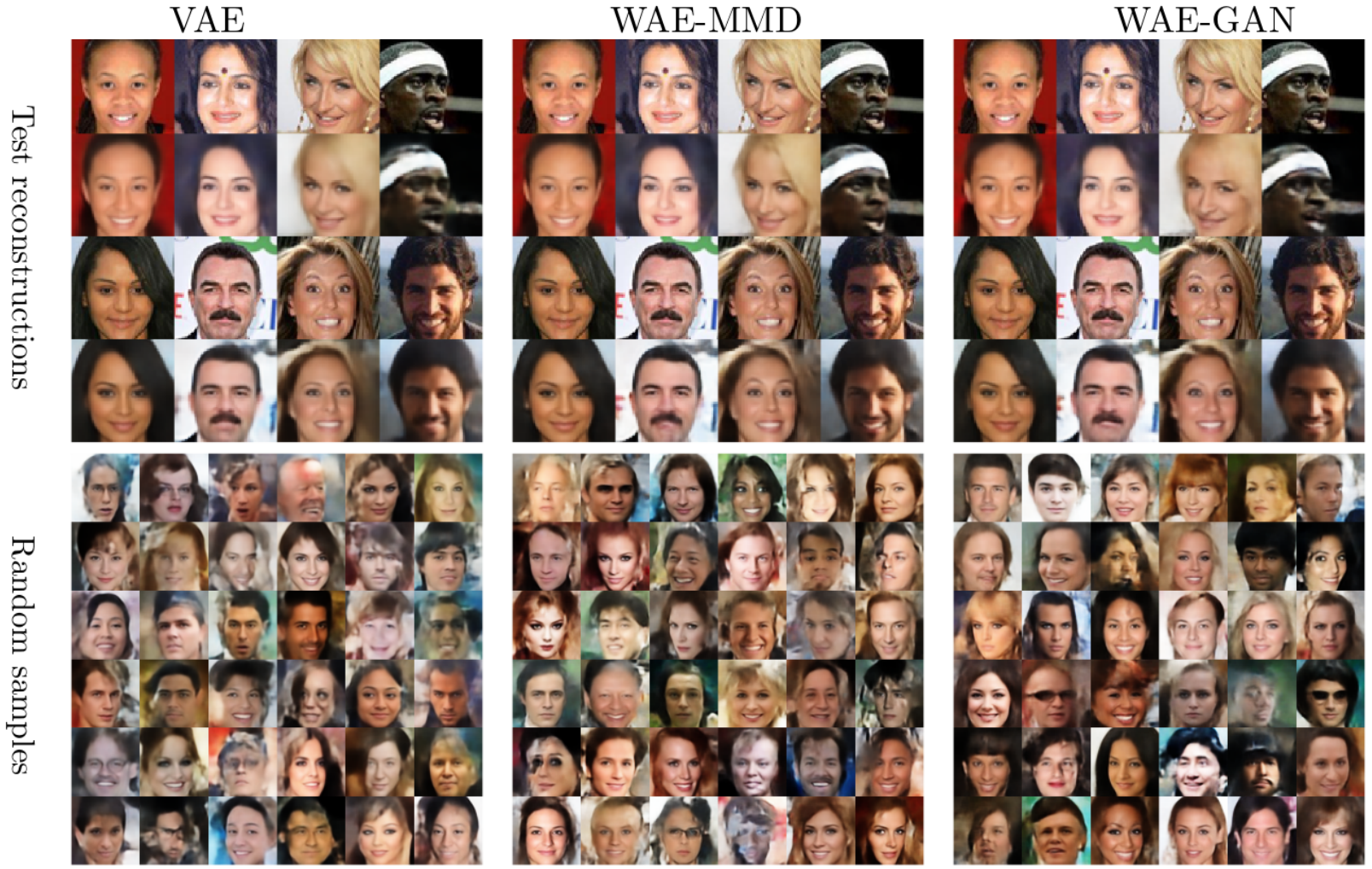
Figure 5: Comparison of the interpolation with L2 Euclidean distance (top), LP Wasserstein interpolation (top middle) regularized Wasserstein Barycenter (down middle) and DWE (down).

Wasserstein Auto-Encoders.
Tolstikhin et al. Dec 2017

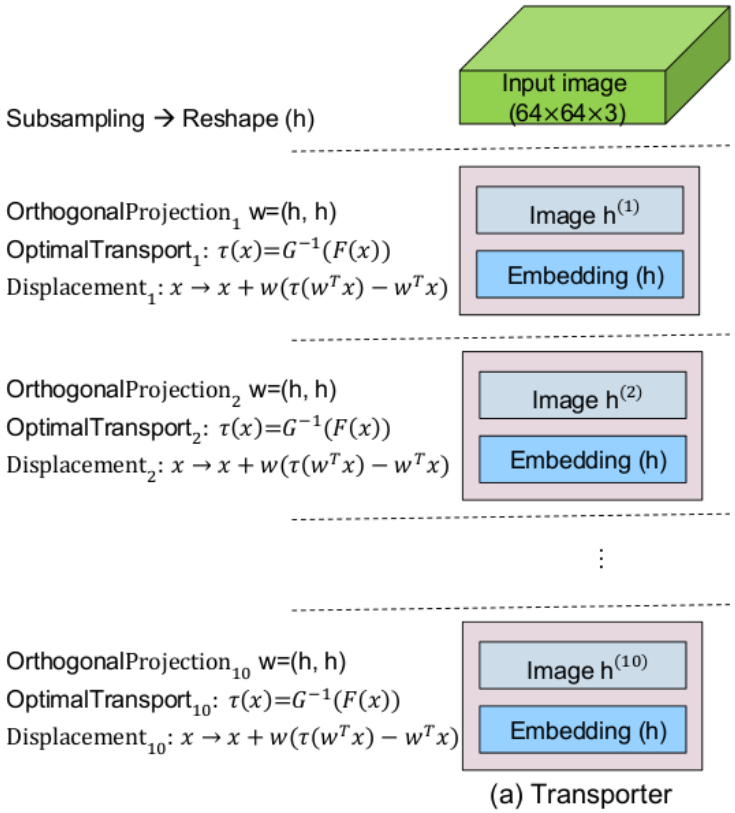
Like Wasserstein GANs, but deriving the regularization instead of imposing a random one.

It is applied to GANs, and to their own Maximum Mean Discrepancy penalty.

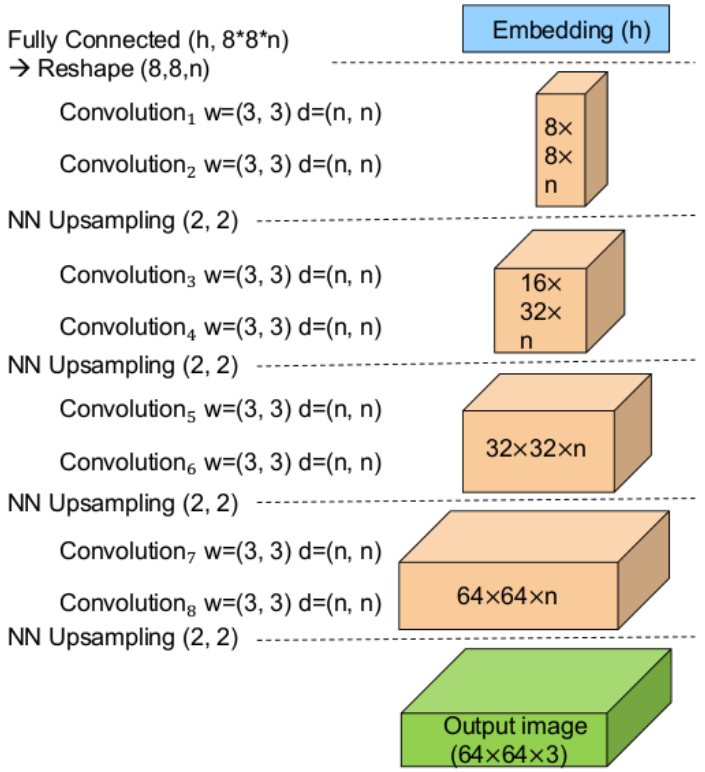



VAE
WAE-MMD
WAE-GAN
Test reconstructions
Random samples

Generative Autotransporters. Wu et al. Nov 2017

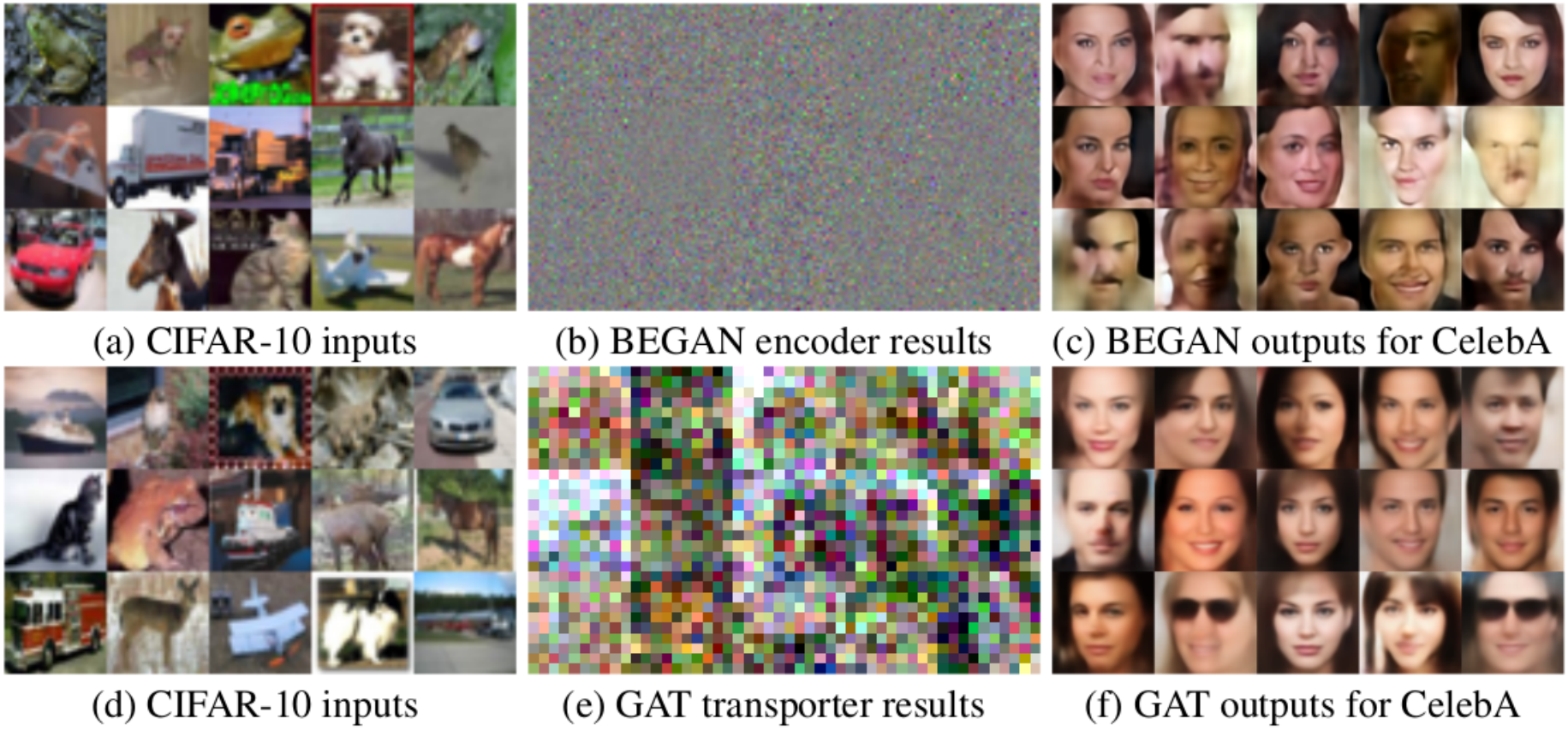


(a) Transporter



(b) Decoder

Generative Autotransporters. Wu et al. Nov 2017



(a) CIFAR-10 inputs (b) BEGAN encoder results (c) BEGAN outputs for CelebA

(d) CIFAR-10 inputs (e) GAT transporter results (f) GAT outputs for CelebA

Figure 5: Results of the BEGAN ((a) → (b) → (c)) and the proposed GAT ((d) → (e) → (f)) with using the CIFAR-10 data as inputs for generating samples that fit the CelebA data.

Earth Mover's Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading (IJCAI-17)

Question	How are overloaded functions differentiated by the compiler?
Model Answer	Based on the function signature. When an overloaded function is called, the compiler will find the function whose signature is closest to the given function call.
Student#1	It looks at the number, types, and order of arguments in the function call
Student#2	By the number, and the types and order of the parameters.

Table 1: Sample question, model answer, and student answers from an undergraduate computer science course [Mohler *et al.*, 2011].

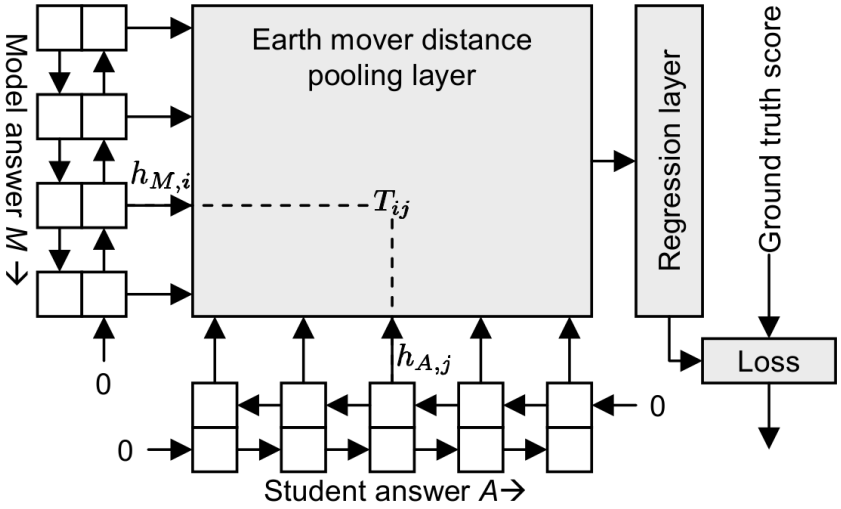


Figure 2: High-level view of our ASAG system.

Earth Mover's Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading (IJCAI-17)

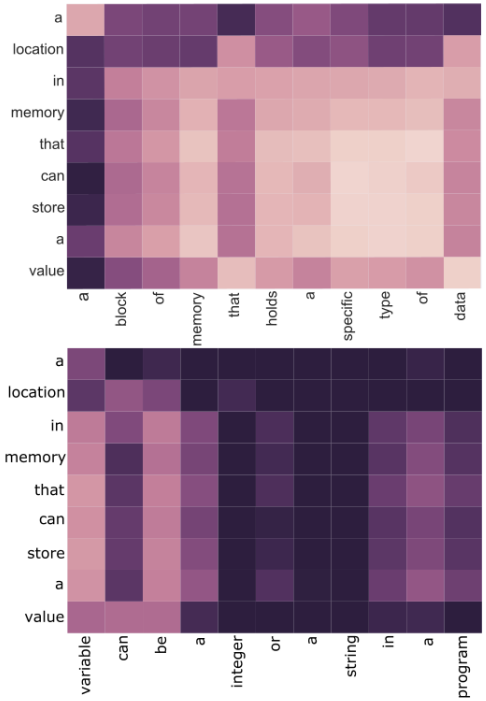


Figure 4: EMD heatmaps for good (score 5/5, above) and poor (score 2/5, below) answers. The good answer shows large values of T_{ij} (bright color) in many cells. The bad answer shows small values (dark colors). Words of M and A are shown along the margins.

	Pooling input	Which state	Pooling	Output stage	System name	MAE	RMSE	Pearson's r
1					Tf-idf		1.022	0.327
2					Lesk		1.050	0.450
3					[Mohler *et al.*, 2011]		0.978	0.518
4	Convnet	All	Avg	SoftMax	ABCNN, [Yin *et al.*, 2015]	0.74	0.92	0.52
5	Convnet	All	Max	SoftMax	[He and Lin, 2016]	0.75	0.87	0.61
6	LSTM	Last		L2	LSTM-Last-L2	0.91	1.101	0.600
7	LSTM	All	EMD	L2	LSTM-EMD-L2	0.96	1.28	0.46
8	LSTM	All	Max	L2	LSTM-MaxPool-L2	1.12	1.60	0.411
9	LSTM	All	Avg	L2	LSTM-AvgPool-L2	1.16	1.58	0.393
10	Word2vec		EMD	SVOR	W2V-EMD-SVOR	0.77	1.073	0.355
11	LSTM	All	Max	SVOR	LSTM-MaxPool-SVOR	0.83	0.973	0.522
12	LSTM	All	Avg	SVOR	LSTM-AvgPool-SVOR	0.63	0.95	0.571
13	LSTM	All	EMD	SVOR	LSTM-EMD-SVOR	**0.490**	**0.830**	0.550
14	LSTM	All	EMD	Logits	LSTM-EMD-Logits	0.657	1.135	**0.649**

Table 3: Performance on Mohler CS dataset with 12-fold training (lower is better for RMSE and MAE; higher is better for Pearson's r). We assess various combinations of input stage, choice of state/s to compare, pooling logic, and regression stage.

Conclusions:

- Earth Mover's Distance is a tricky Loss to use but:
  - It has become way more available recently.
  - It allows us to Embed a Metric from the Output Space.
  - It is Natural for Histogram Prediction / Probability Distributions, etc.

- Resources:
  - Math: http://marcocuturi.net/SI.html
  - Deep Learning: http://cbcl.mit.edu/wasserstein/
  - Book: https://optimaltransport.github.io/
  - Fancy but broken uses:
    - Wasserstein GAN
    - Improved Training of Wasserstein GANs
    - Improving the Improved Training of Wasserstein GANs

References

- "Sinkhorn Distances: Lightspeed Computation of Optimal Transport", M Cuturi, NIPS 2013
- "Learning with a Wasserstein loss", Frogner et Al. NIPS 2015

- "Stochastic optimization for large-scale optimal transport", Genevay et Al. NIPS 2016

- "Supervised word mover's distance", Huang et Al. NIPS 2016

- "Joint distribution optimal transportation for domain adaptation", Courty et Al. NIPS 2017
- "Learning Wasserstein Embeddings", Courty et Al. Arxiv, 2017 (october)
- "Generative Autotransporters", Wu et Al. Arxiv, 2017 (november), "Wasserstein Auto-Encoders", Tolstikhin et Al. Arxiv 2017 (december)
- "Earth Mover's Distance Pooling over Siamese LSTMs for Automatic Short Answer Grading", Kumar et Al. IJCAI 2017

Thanks for your Attention!
Questions?