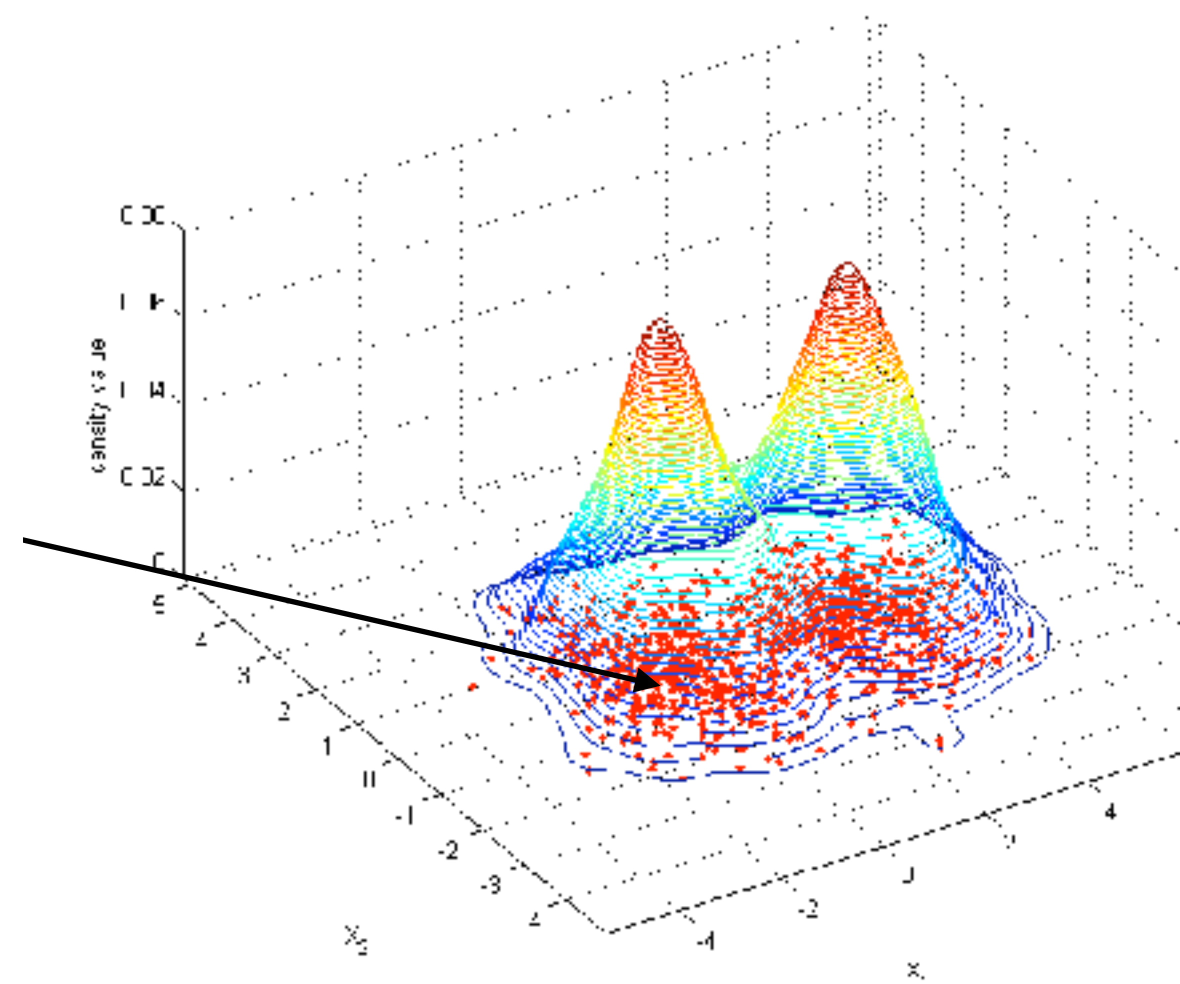
Applications of **optimal transport** to machine learning and signal processing

Présentation par **Nicolas Courty**
Maître de conférences HDR / Université de Bretagne Sud
Laboratoire IRISA
http://people.irisa.fr/Nicolas.Courty/

Motivations

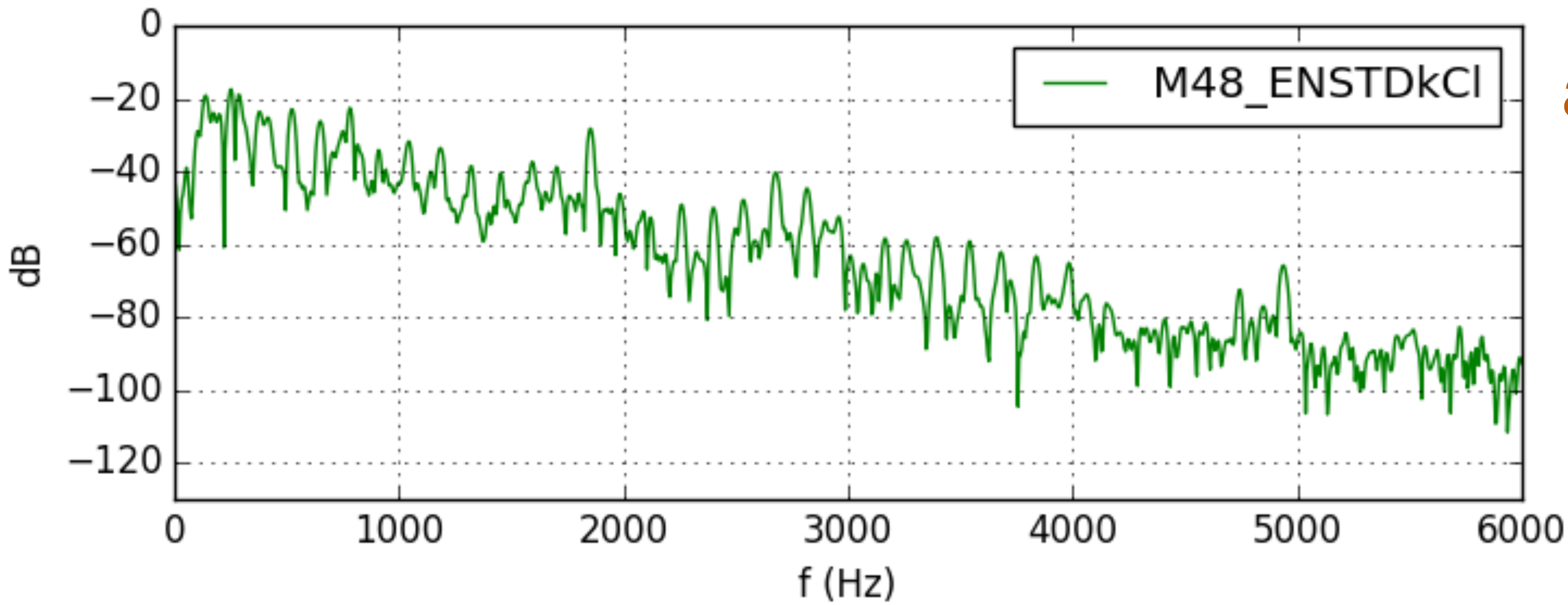
- Optimal transport is a perfect tool to compare empirical probability distributions
- In the context of machine learning/signal processing, one often has to deal with collections of samples that can be interpreted as probability distributions



Motivations

- Optimal transport is a perfect tool to compare empirical probability distributions

- In the context of machine learning/signal processing, one often has to deal with collections of samples that can be interpreted as probability distributions



a piano note

with proper normalization: probability distribution !

Motivations

- I will showcase 2 successful examples of application of OT in the contexte of machine learning and signal processing
- **First one: OT for transfer learning (domain adaptation)**
  - using the coupling to interpolate multidimensional data
  - special note on the out-of-sample problem
- **Second: OT for music transcription**
  - using the metric to adapt to the specificity of the data

Forenote on implementation

- All these examples have been implemented using

  POT, the Python Optimal Transport toolbox

- Available here : https://github.com/rflamary/POT

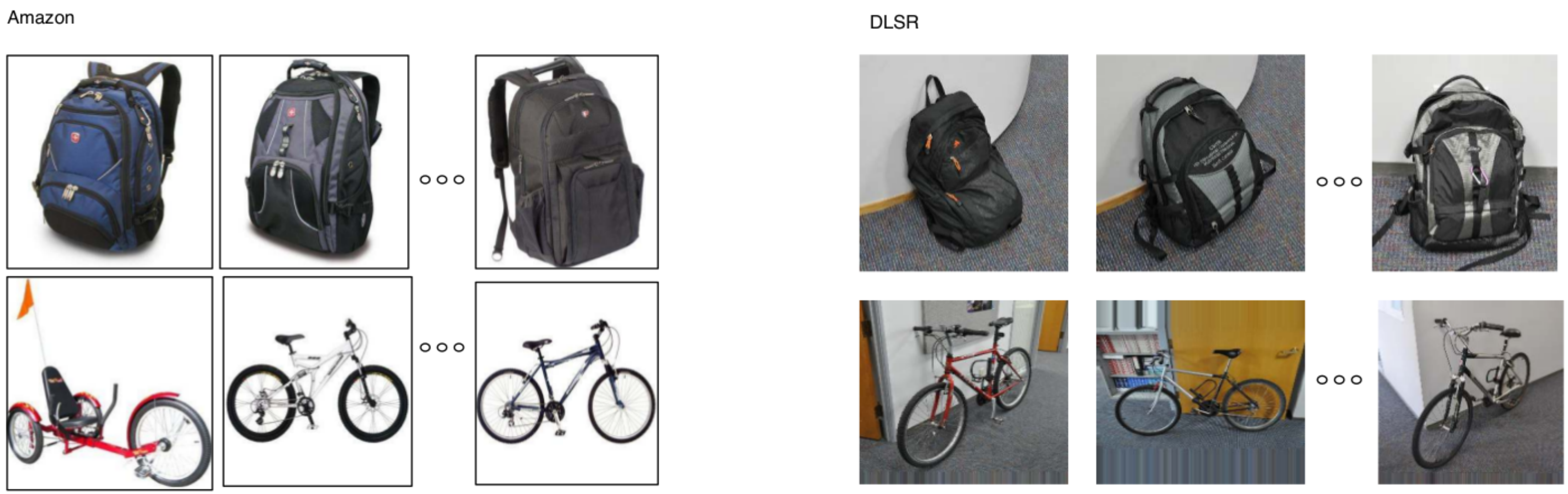
- Some use cases will be given along the examples

Optimal Transport for domain adaptation

introduction to domain adaptation
regularization helps
out of samples formulation

Joint work with Rémi Flamary, Devis Tuia, Alain Rakotomamonjy, Michael Perrot, Amaury Habrard

Domain Adaptation problem



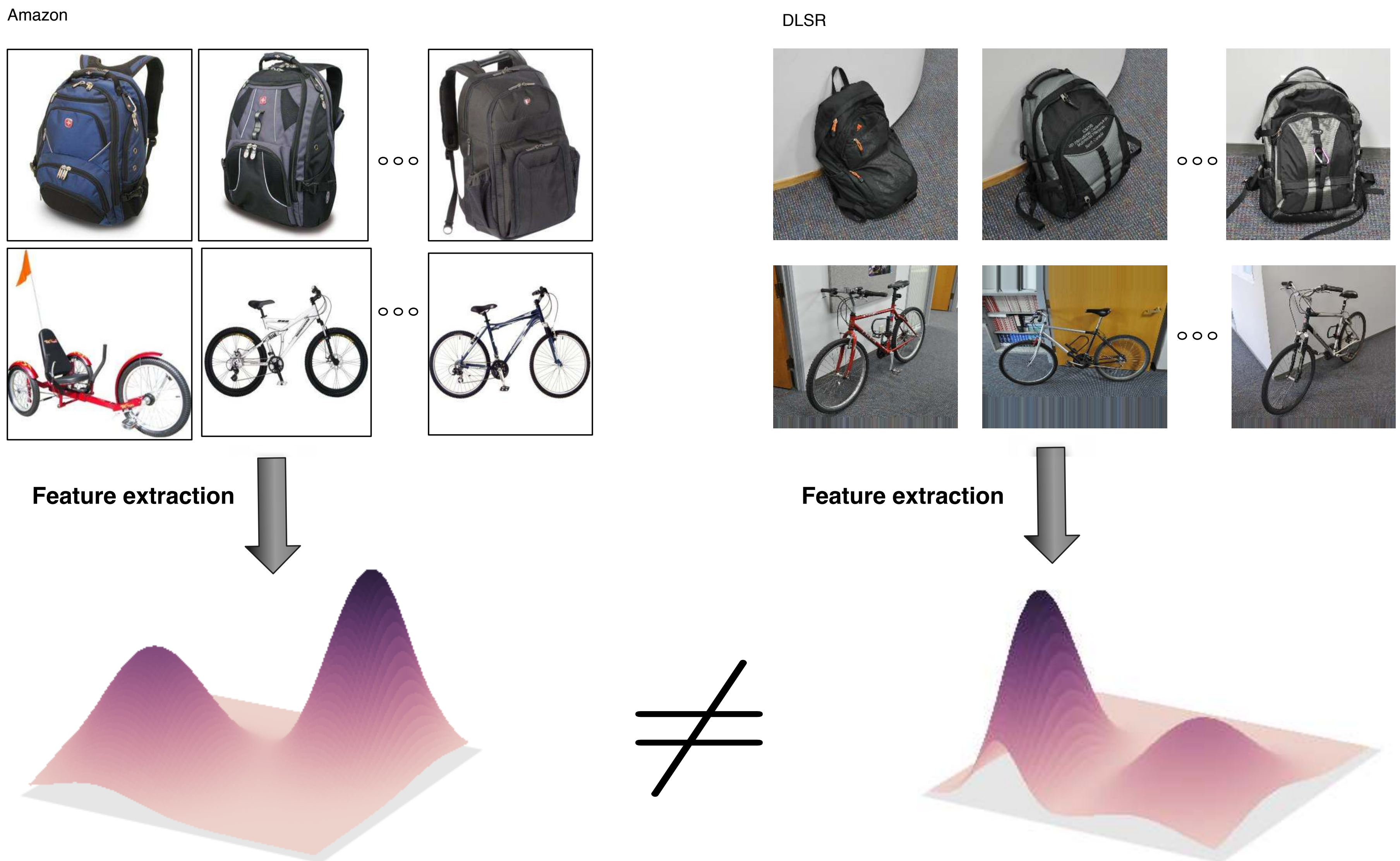
Traditional machine learning hypothesis

- We have access to training data.
- Probability distribution of the training set and the testing are the same.
- We want to learn a classifier that generalizes to new data.

Our context

- Classification problem with data coming from different sources (domains).
- Distributions are different but related.

Domain Adaptation problem

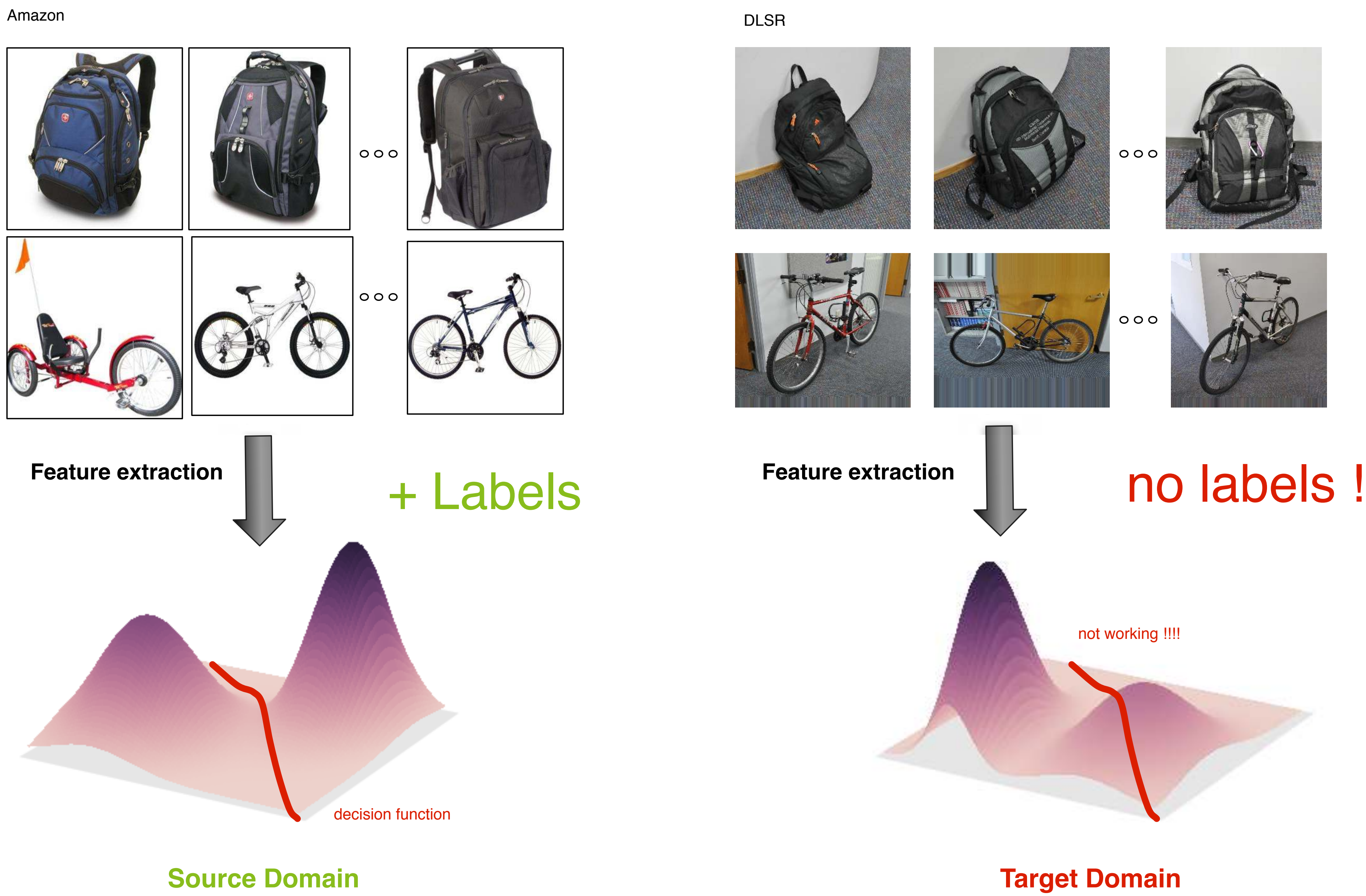


Probability Distribution Functions over the domains

Our context

- Classification problem with data coming from different sources (domains).
- Distributions are different but related.

Unsupervised domain adaptation problem



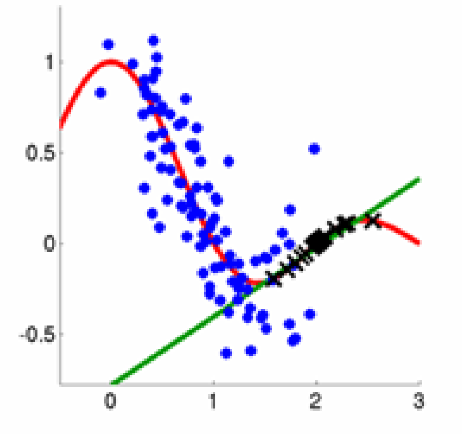
Problems

- Labels only available in the **source domain**, and classification is conducted in the **target domain**.
- Classifier trained on the source domain data performs badly in the target domain

Domain adaptation short state of the art

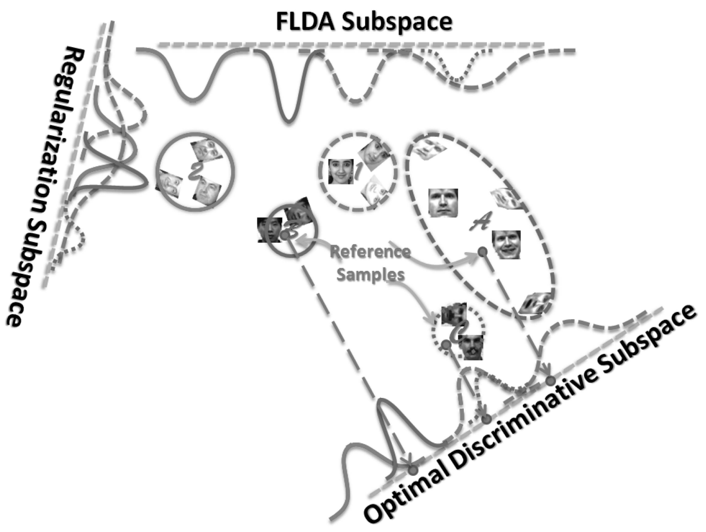
Reweighting schemes [Sugiyama et al., 2008]

- Distribution change between domains.
- Reweigh samples to compensate this change.



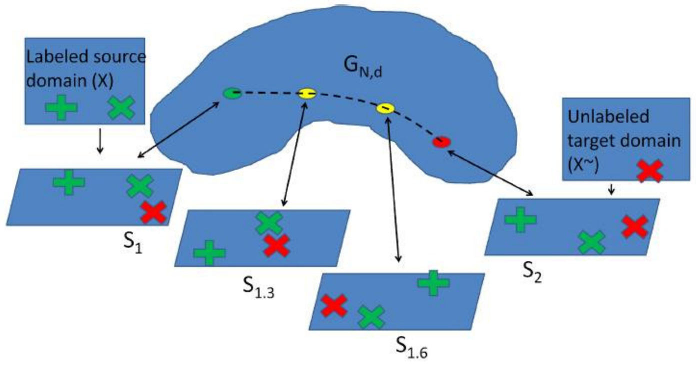
Subspace methods

- Data is invariant in a common latent subspace.
- Minimization of a divergence between the projected domains [Si et al., 2010].
- Use additional label information [Long et al., 2014].



Gradual alignment

- Alignment along the geodesic between source and target subspace [R. Gopalan and Chellappa, 2014].
- Geodesic flow kernel [Gong et al., 2012].



Generalization error in domain adaptation

Theoretical bounds [Ben-David et al., 2010]

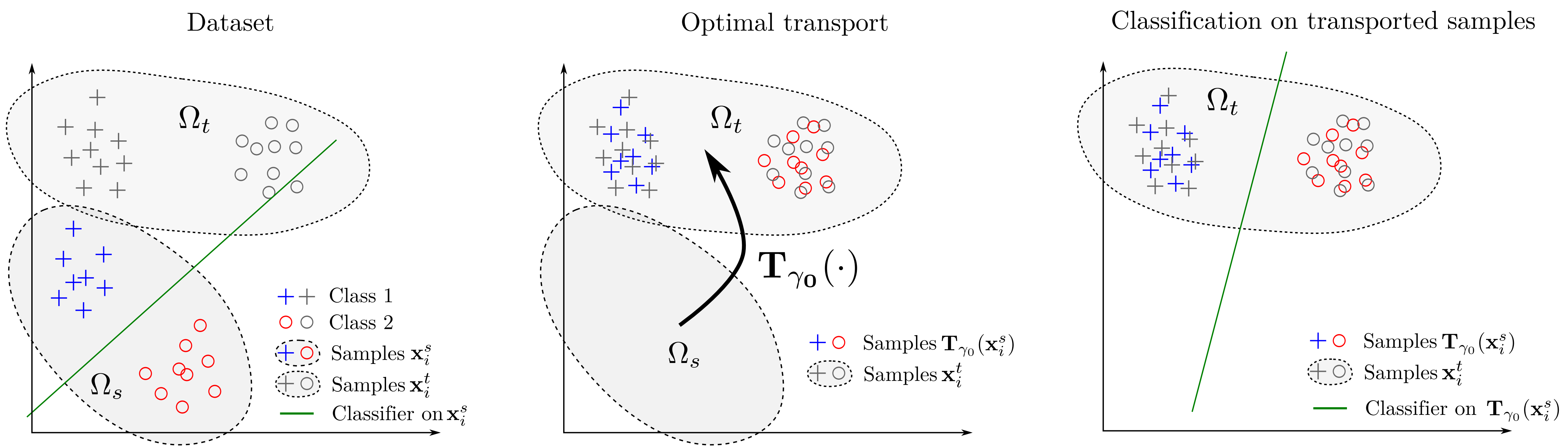
The error performed by a given classifier in the target domain is upper-bounded by the sum of three terms :

- Error of the classifier in the source domain;
- Divergence measure between the two pdfs in the two domains;
- A third term measuring how much the classification tasks are related to each other.

Our proposal [Courty et al., 2016]

- Model the discrepancy between the distribution through a general transformation.
- Use **optimal transport** to estimate the transportation map between the two distributions.
- Use regularization terms for the optimal transport problem that exploits labels from the source domain.

Optimal transport for domain adaptation



Assumptions

- There exist a transport $\mathbf{T}$ between the source and target domain.
- The transport preserves the conditional distributions:

$$P_s(y|\mathbf{x}_s) = P_t(y|\mathbf{T}(\mathbf{x}_s)).$$

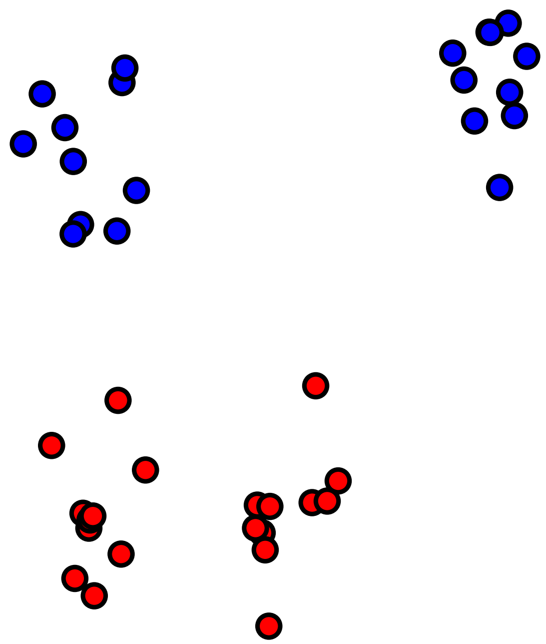
3-step strategy

1. Estimate optimal transport between distributions.
2. Transport the training samples onto the target distribution.
3. Learn a classifier on the transported training samples.

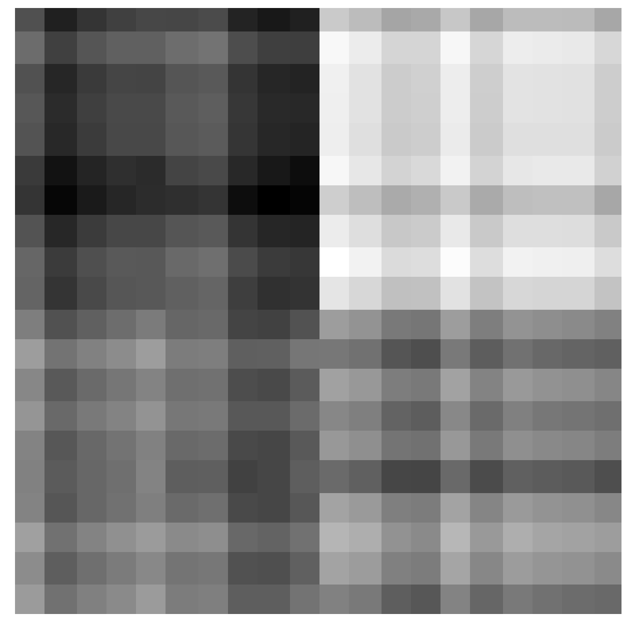
Optimal Transport for domain adaptation

introduction to domain adaptation
regularization helps
out of samples formulation

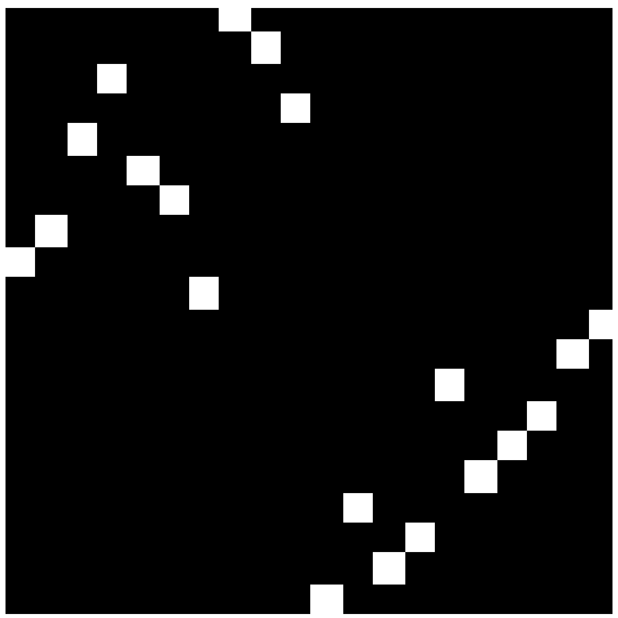
Optimal transport for empirical distributions



Transportation cost matric $\mathbf{C}$



Optimal matrix γ (LP)

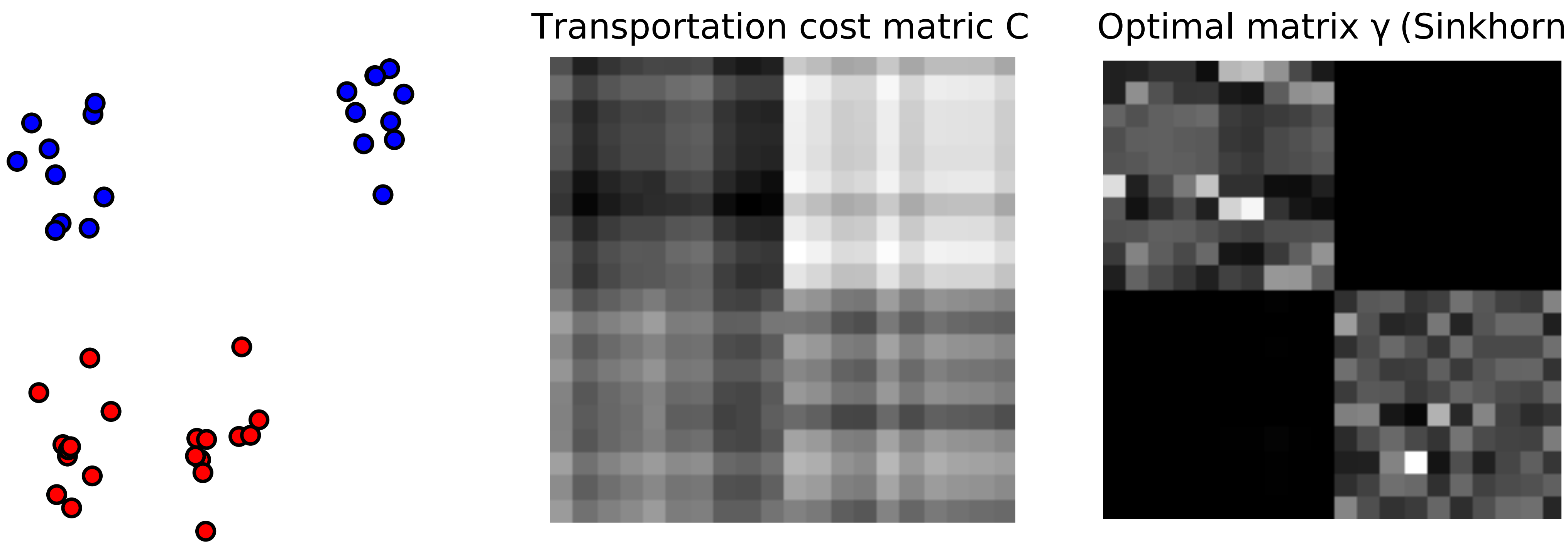


Empirical distributions

$$\mu_s = \sum_{i=1}^{n_s} p_i^s \delta_{\mathbf{x}_i^s}, \quad \mu_t = \sum_{i=1}^{n_t} p_i^t \delta_{\mathbf{x}_i^t} \tag{4}$$

- $\delta_{\mathbf{x}_i}$ is the Dirac at location $\mathbf{x}_i \in \mathbb{R}^d$ and p_i^s and p_i^t are probability masses.
- $\sum_{i=1}^{n_s} p_i^s = \sum_{i=1}^{n_t} p_i^t = 1$, in this work $p_i^s = \frac{1}{n_s}$ and $p_i^t = \frac{1}{n_t}$.
- Samples stored in matrices: $\mathbf{X}_s = [\mathbf{x}_1^s, \ldots, \mathbf{x}_{ns}^s]^\top$ and $\mathbf{X}_t = [\mathbf{x}_1^t, \ldots, \mathbf{x}_{nt}^t]^\top$
- The cost is set to the squared Euclidean distance $C_{i,j} = \|\mathbf{x}_i^s - \mathbf{x}_j^t\|^2$.
- Same optimization problem, different $\mathbf{C}$.

Efficient regularized optimal transport



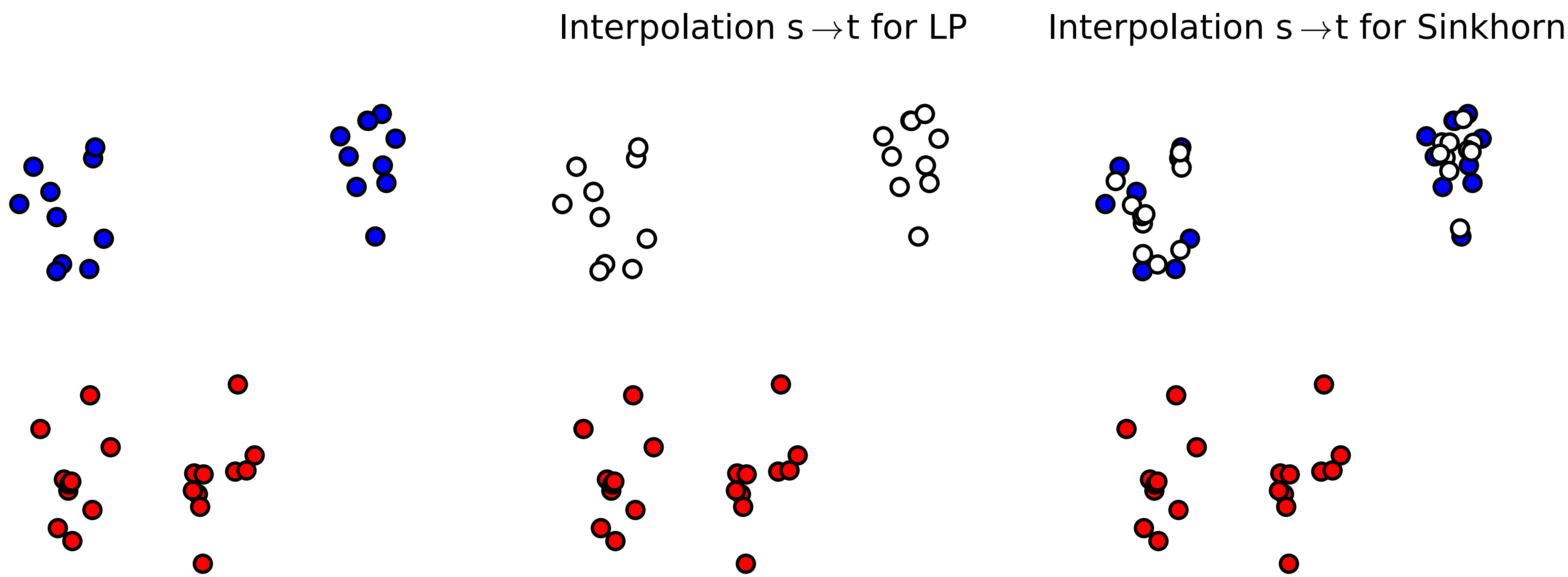
Entropic regularization [Cuturi, 2013]

$$\gamma_0^\lambda = \underset{\gamma \in \mathcal{P}}{\arg\min} \, \langle \gamma, \mathbf{C} \rangle_F - \lambda h(\gamma), \tag{5}$$

where $h(\gamma) = -\sum_{i,j} \gamma(i,j) \log \gamma(i,j)$ computes the entropy of γ.

- Entropy introduces smoothness in γ_0^λ.
- **Sinkhorn-Knopp** algorithm (efficient implementation in parallel, GPU).
- General framework using Bregman projections [Benamou et al., 2015].

Transporting the discrete samples



Barycentric mapping [Ferradans et al., 2014]

- The mass of each source sample is spread onto the target samples (line of $\boldsymbol{\gamma}_0$).
- The source samples becomes a weighted sum of dirac (impractical for ML).
- We estimate the transported position for each source with:

$$\widehat{\mathbf{x}_i^s} = \arg\min_{\mathbf{x}} \quad \sum_j \boldsymbol{\gamma}_0(i,j) c(\mathbf{x}, \mathbf{x}_j^t). \tag{6}$$

- Position of the transported samples for squared Euclidean loss:

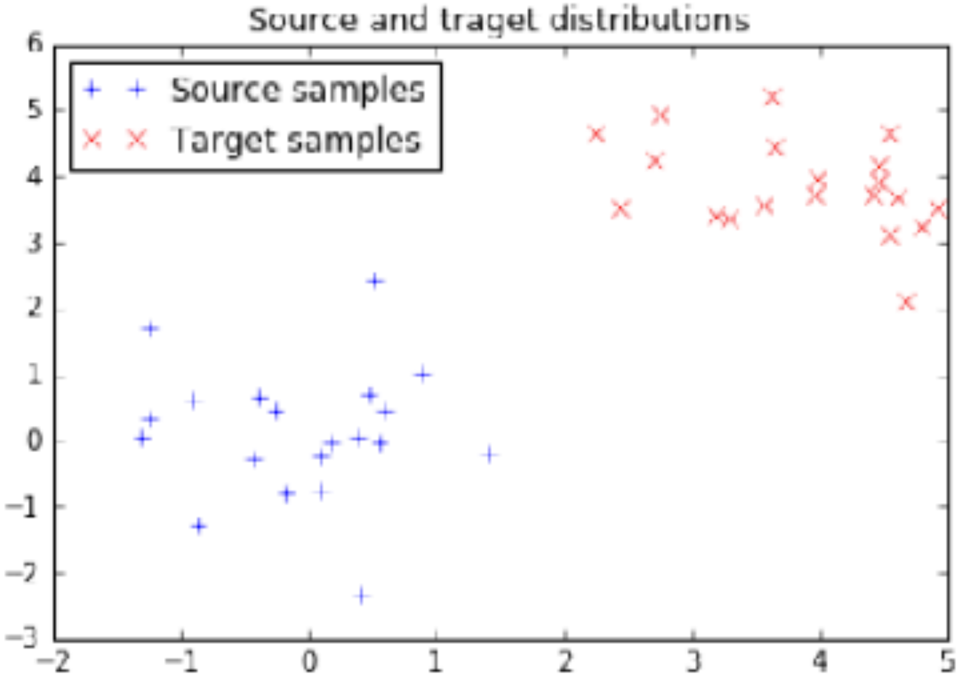
$$\hat{\mathbf{X}}_s = \text{diag}(\boldsymbol{\gamma}_0 \mathbf{1}_{n_t})^{-1} \boldsymbol{\gamma}_0 \mathbf{X}_t \quad \text{and} \quad \hat{\mathbf{X}}_t = \text{diag}(\boldsymbol{\gamma}_0^\top \mathbf{1}_{n_s})^{-1} \boldsymbol{\gamma}_0^\top \mathbf{X}_s. \tag{7}$$

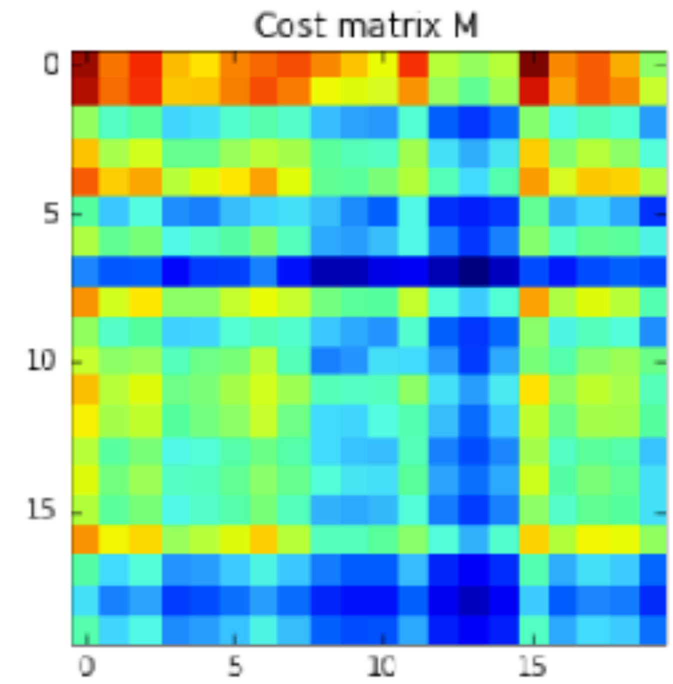
In POT

```
In [1]: import numpy as np
        import matplotlib.pylab as pl
        import ot
```

0.1 Data generation

```
In [2]: n=20 # nb samples

        mu_s=np.array([0,0])
        cov_s=np.array([[1,0],[0,1]])

        mu_t=np.array([4,4])
        cov_t=np.array([[1,-.8],[-.8,1]])

        xs=ot.datasets.get_2D_samples_gauss(n,mu_s,cov_s)
        xt=ot.datasets.get_2D_samples_gauss(n,mu_t,cov_t)

        a,b = ot.unif(n),ot.unif(n) # uniform distribution on samples

        # loss matrix
        M=ot.dist(xs,xt)
        M/=M.max()
```





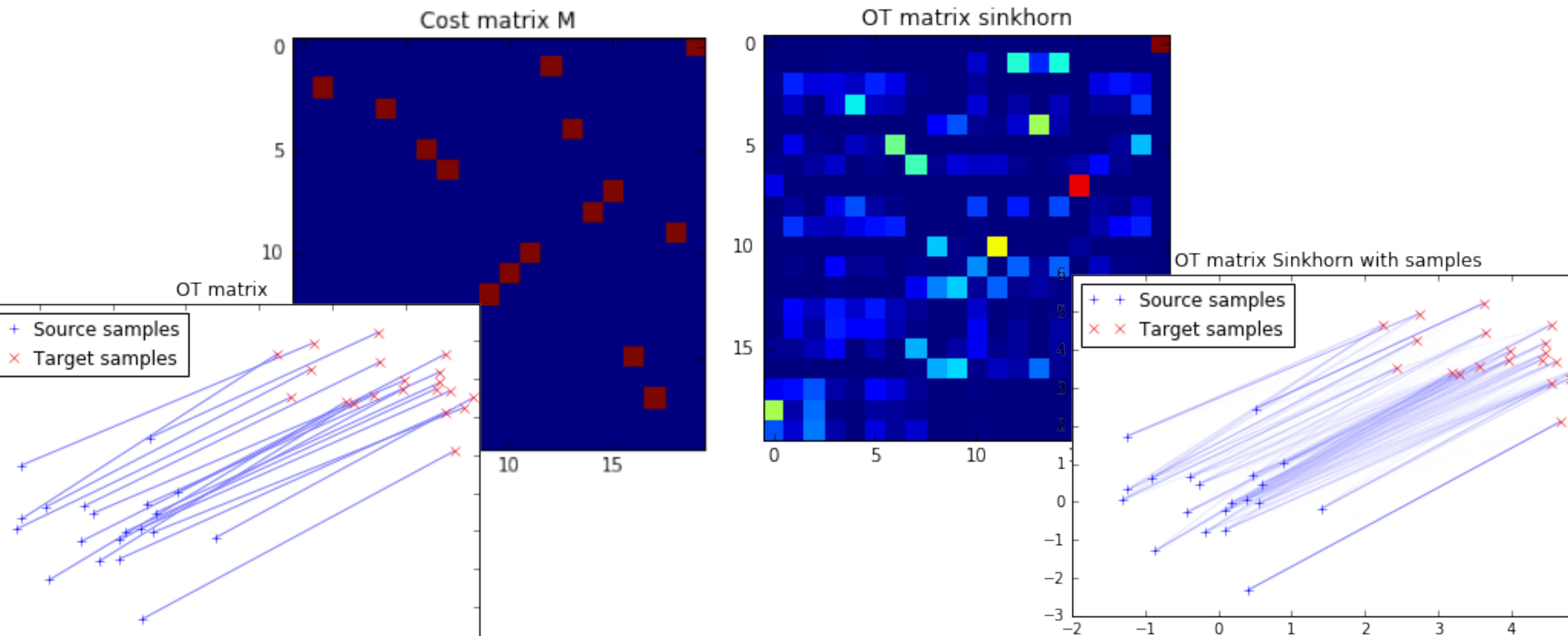
In POT

LP

```
In [4]: G0=ot.emd(a,b,M)


        pl.figure(3)
        pl.imshow(G0,interpolation='nearest')
        pl.title('Cost matrix M')

        pl.figure(4)
        ot.plot.plot2D_samples_mat(xs,xt,G0,c=[.5,.5,1])
        pl.plot(xs[:,0],xs[:,1],'+b',label='Source samples')
        pl.plot(xt[:,0],xt[:,1],'xr',label='Target samples')
        pl.legend(loc=0)
        pl.title('OT matrix')

Out[4]: <matplotlib.text.Text at 0x7f4fa724b150>
```

Sinkhorn

```
In [5]: # reg term
        lambd=5e-3

        Gs=ot.sinkhorn(a,b,M,lambd)

        pl.figure(5)
        pl.imshow(Gs,interpolation='nearest')
        pl.title('OT matrix sinkhorn')

        pl.figure(6)
        ot.plot.plot2D_samples_mat(xs,xt,Gs,color=[.5,.5,1])
        pl.plot(xs[:,0],xs[:,1],'+b',label='Source samples')
        pl.plot(xt[:,0],xt[:,1],'xr',label='Target samples')
        pl.legend(loc=0)
        pl.title('OT matrix Sinkhorn with samples')

        <matplotlib.text.Text at 0x7f4fa703c550>
```



Regularization for domain adaptation

Optimization problem

$$\min_{\boldsymbol{\gamma}\in\mathcal{P}} \quad \langle \boldsymbol{\gamma}, \mathbf{C} \rangle_F + \lambda\Omega_s(\boldsymbol{\gamma}) + \eta\Omega(\boldsymbol{\gamma}), \tag{8}$$

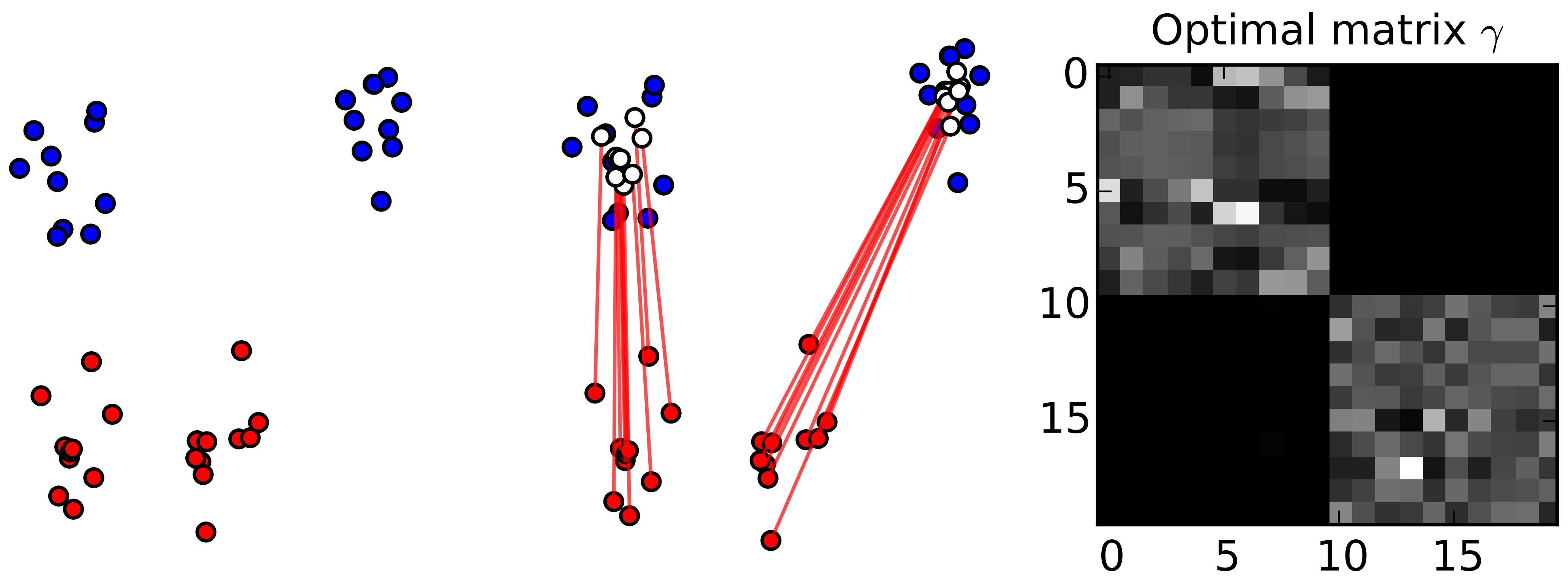
where

- $\Omega_s(\boldsymbol{\gamma})$ Entropic regularization [Cuturi, 2013].
- $\eta \geq 0$ and $\Omega_c(\cdot)$ is a DA regularization term.
- Regularization to avoid overfitting in high dimension and encode additional information.

Regularization terms for domain adaptation $\Omega(\boldsymbol{\gamma})$

- Class based regularization [Courty et al., 2014] to encode the source label information.
- Graph regularization [Ferradans et al., 2014] to promote local sample similarity conservation.
- Semi-supervised regularization when some target samples have known labels.

Entropic regularization

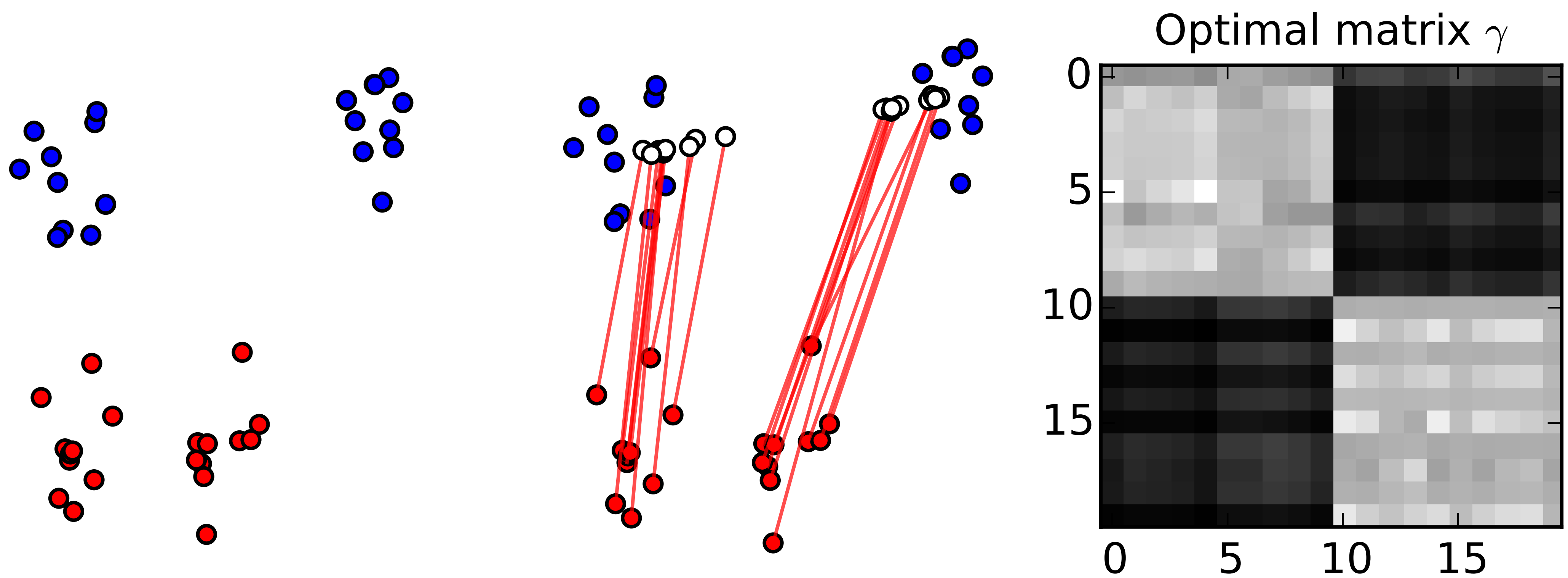


Entropic regularization [Cuturi, 2013]

$$\Omega_s(\boldsymbol{\gamma}) = \sum_{i,j} \boldsymbol{\gamma}(i,j) \log \boldsymbol{\gamma}(i,j)$$

- Extremely efficient optimization scheme (Sinkhorn Knopp).
- Solution is not sparse anymore due to the regularization.
- Strong regularization force the samples to concentrate on the center of mass of the target samples.

Entropic regularization

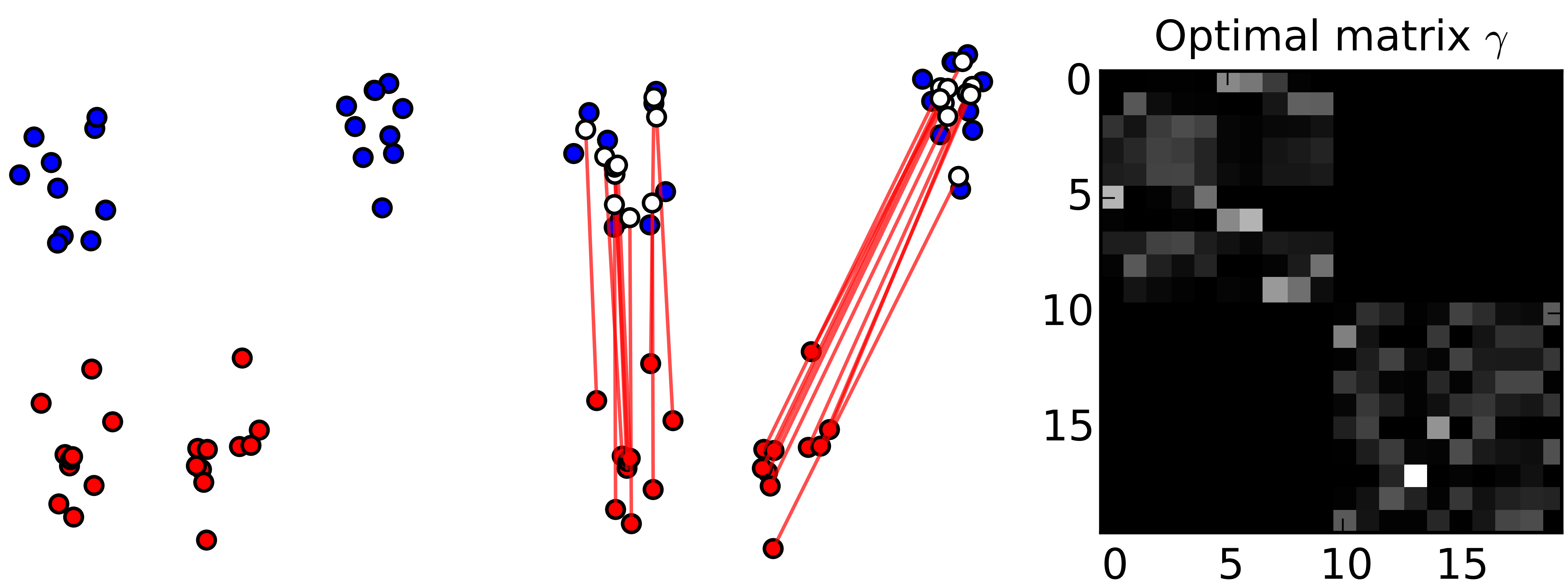


Entropic regularization [Cuturi, 2013]

$$\Omega_s(\boldsymbol{\gamma}) = \sum_{i,j} \boldsymbol{\gamma}(i,j) \log \boldsymbol{\gamma}(i,j)$$

- Extremely efficient optimization scheme (Sinkhorn Knopp).
- Solution is not sparse anymore due to the regularization.
- Strong regularization force the samples to concentrate on the center of mass of the target samples.

Class-based regularization



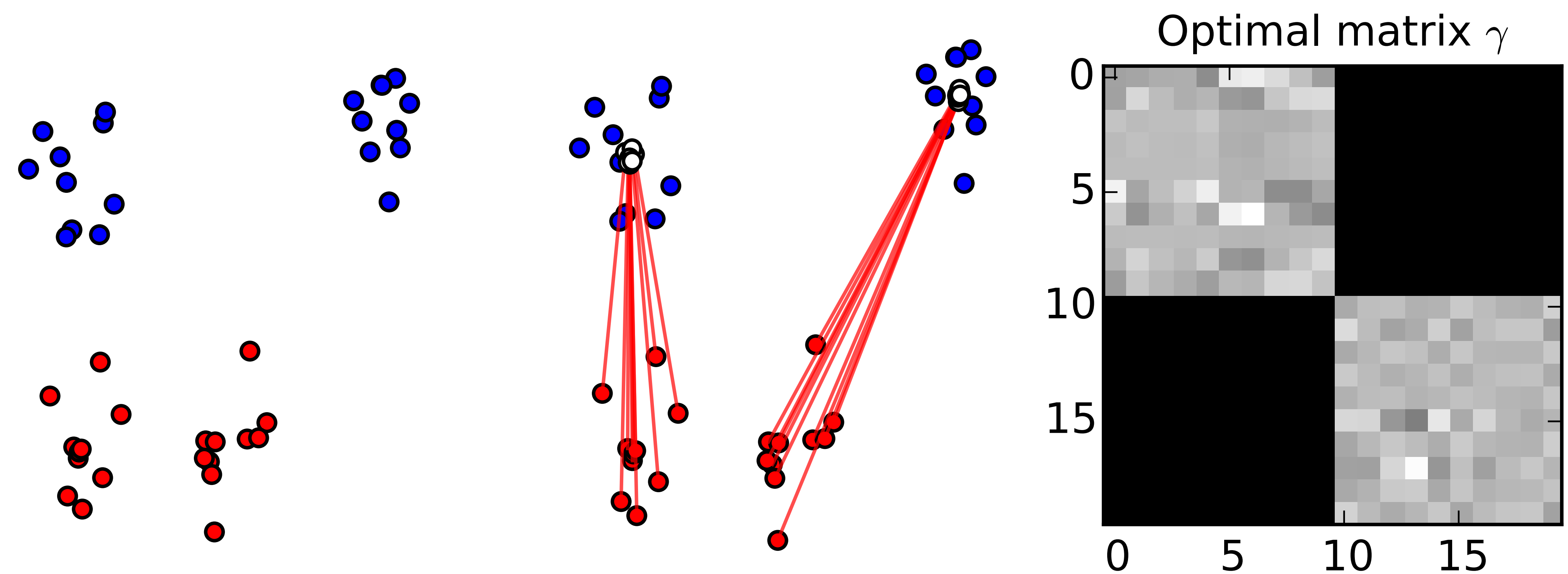
Group lasso regularization [Courty et al., 2016]

- We group components of $\boldsymbol{\gamma}$ using classes from the source domain:

$$\Omega_c(\boldsymbol{\gamma}) = \sum_j \sum_c ||\boldsymbol{\gamma}(\mathcal{I}_c, j)||_q^p, \tag{9}$$

- $\mathcal{I}_c$ contains the indices of the lines related to samples of the class c in the source domain.
- $|| \cdot ||_q^p$ denotes the ℓ_q norm to the power of p.
- For $p \leq 1$, we encourage a target domain sample j to receive masses only from "same class" source samples.

Class-based regularization



Group lasso regularization [Courty et al., 2016]

- We group components of $\boldsymbol{\gamma}$ using classes from the source domain:

$$\Omega_c(\boldsymbol{\gamma}) = \sum_j \sum_c ||\boldsymbol{\gamma}(\mathcal{I}_c, j)||_q^p, \tag{9}$$

- $\mathcal{I}_c$ contains the indices of the lines related to samples of the class c in the source domain.
- $||\cdot||_q^p$ denotes the ℓ_q norm to the power of p.
- For $p \leq 1$, we encourage a target domain sample j to receive masses only from "same class" source samples.

Optimization problem

$$\min_{\boldsymbol{\gamma}\in\mathcal{P}} \quad \langle \boldsymbol{\gamma}, \mathbf{C}\rangle_F + \lambda\Omega_s(\boldsymbol{\gamma}) + \eta\Omega(\boldsymbol{\gamma}),$$

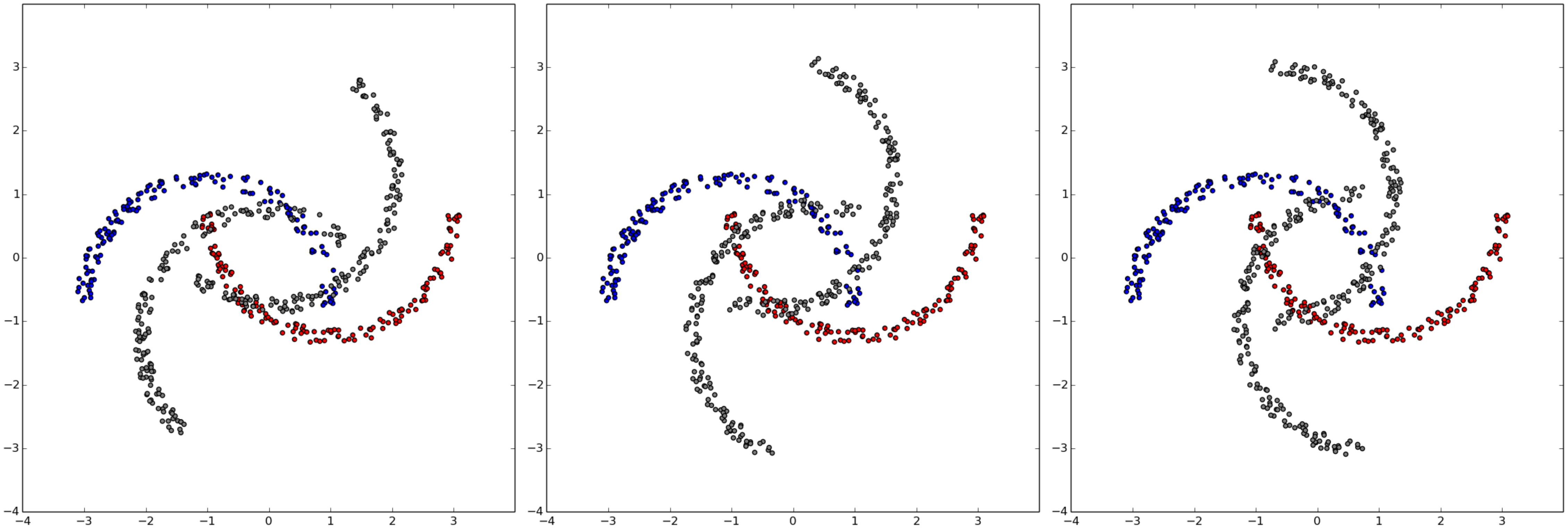
Special cases

- $\eta = 0$: Sinkhorn Knopp [Cuturi, 2013].
- $\lambda = 0$ and Laplacian regularization: Large quadratic program solved with conditionnal gradient [Ferradans et al., 2014].
- Non convex group lasso $\ell_p - \ell_1$: Majoration Minimization with Sinkhorn Knopp [Courty et al., 2014].

General framework with convex regularization $\Omega(\gamma)$

- Can we use efficient Sinkhorn Knopp scaling to solve the global problem?
- Yes using generalized conditional gradient [Bredies et al., 2009].
- Linearization of the second regularization term but not the entropic regularization.

Simulated problem with controllable complexity



Two moons problem [Germain et al., 2013]

- Two entangled moons with a rotation between domains.
- The rotation angle allow a control of the adaptation difficulty.
- Comparison with Domain Adaptation SVM[Bruzzone and Marconcini, 2010] and [Germain et al., 2013].

OT domain adaptation:

- **OT-exact** non-regularized OT.
- **OT-IT** Entropic reg.
- **OT-GL** Group-lasso + entropic reg.
- **OT-Lap** Laplacian + entropic reg.

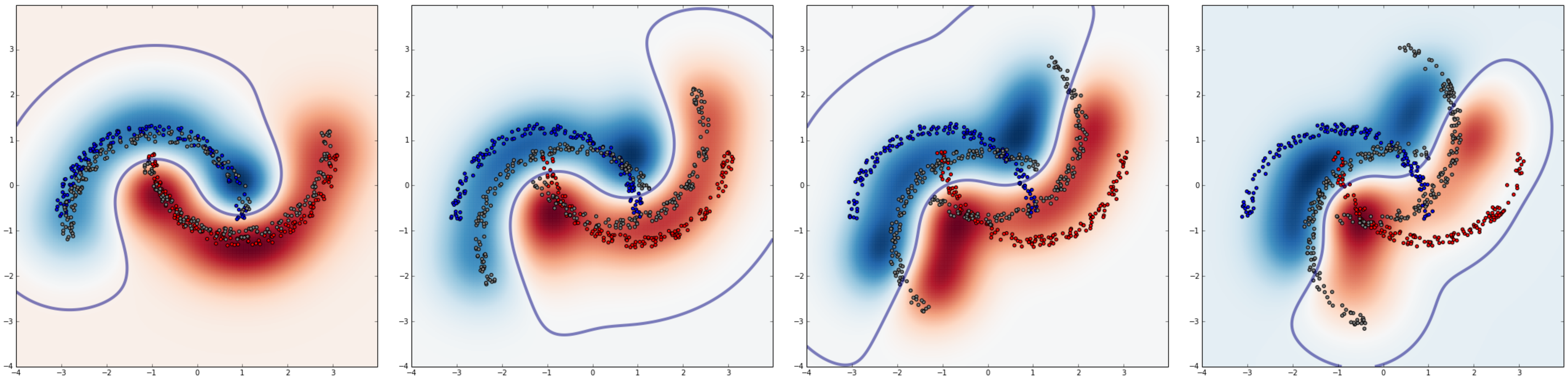
Results on the two moons dataset

	10°	20°	30°	40°	50°	70°	90°
SVM (no adapt.)	0	0.104	0.24	0.312	0.4	0.764	0.828
DASVM	0	**0**	0.259	0.284	0.334	0.747	0.820
PBDA	0	0.094	0.103	0.225	0.412	0.626	0.687
OT-exact	0	0.028	0.065	0.109	0.206	0.394	**0.507**
OT-IT	0	0.007	0.054	0.102	0.221	0.398	**0.508**
OT-GL	0	**0**	**0**	**0.013**	**0.196**	**0.378**	**0.508**
OT-Lap	0	**0**	0.004	0.062	0.201	0.402	0.524

Discussion

- Average prediction error for adaptation from 10° to 90°.
- Clear advantage of the optimal transport techniques.
- Regularization helps (a lot) up to 40°.
- 90° is the theoretical limit (positive definite Jacobian of the transformation).

Results on the two moons dataset

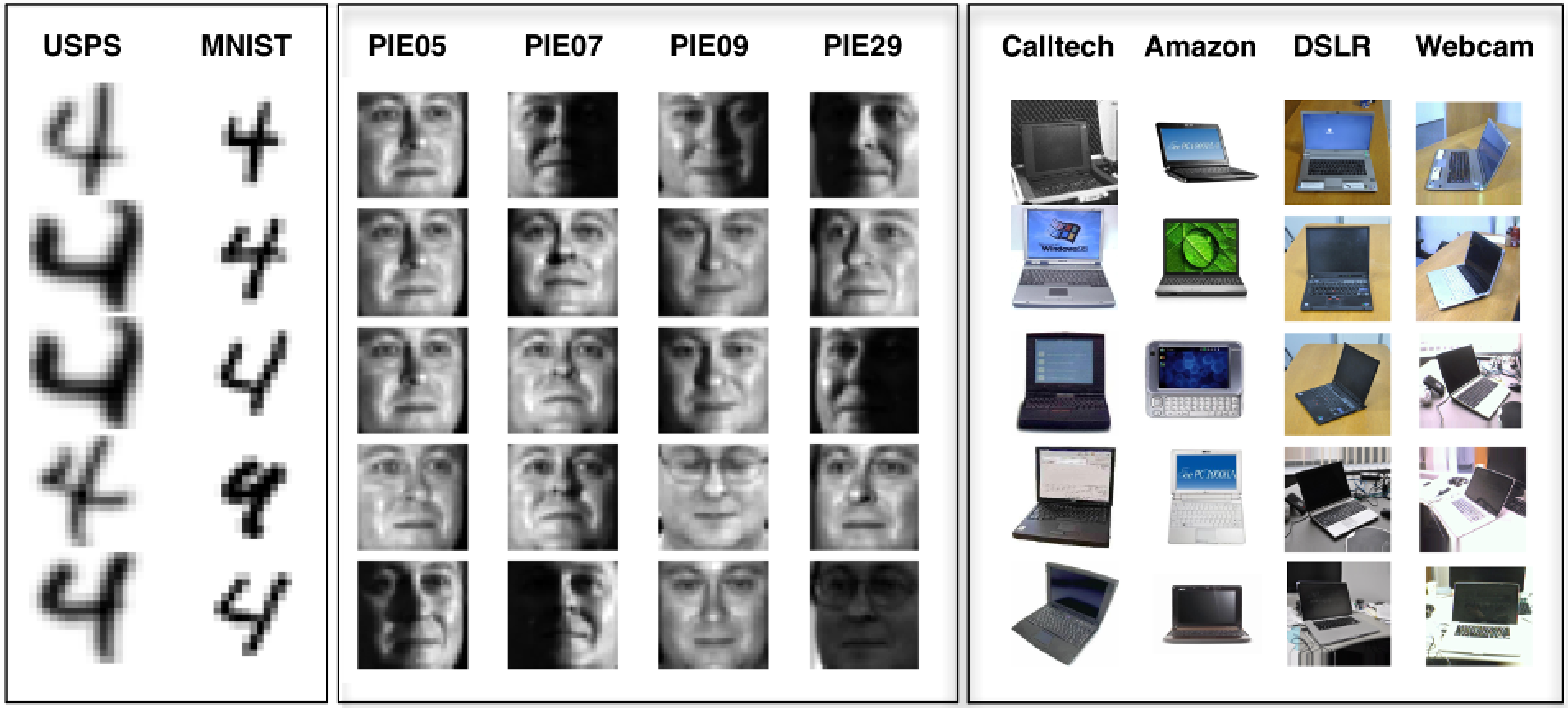


(a) rotation=10° (b) rotation=30° (c) rotation=50° (d) rotation=70°

Discussion

- Average prediction error for adaptation from 10° to 90°.
- Clear advantage of the optimal transport techniques.
- Regularization helps (a lot) up to 40°.
- 90° is the theoretical limit (positive definite Jacobian of the transformation).

Visual adaptation datasets



Datasets

- **Digit recognition**, MNIST VS USPS (10 classes, d=256, 2 dom.).
- **Face recognition**, PIE Dataset (68 classes, d=1024, 4 dom.).
- **Object recognition**, Caltech-Office dataset (10 classes, d=800/4096, 4 dom.).

Numerical experiments

- Comparison with state of the art on the 3 datasets.
- Comparison on object recognition with deep invariant features.
- Semi supervised extension.

Comparison on vision datasets

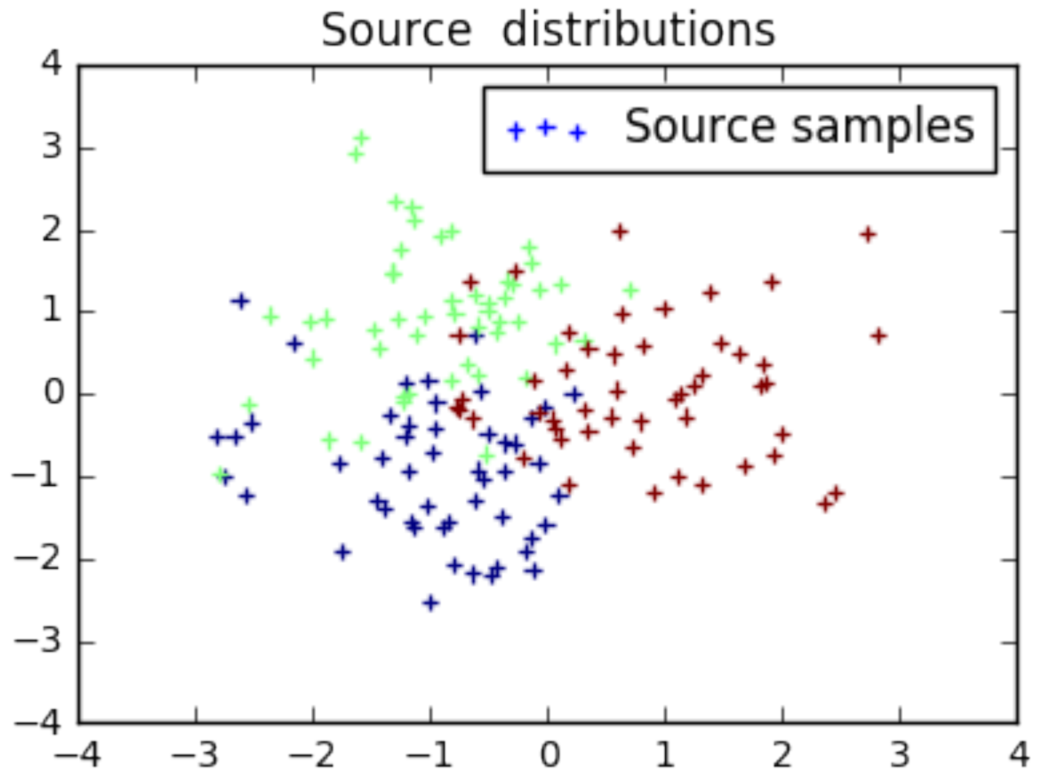
Datasets / Methods	Digits ACC	Digits Nb best	Faces ACC	Faces Nb best	Objects ACC	Objects Nb best
1NN	48.66	0	26.22	0	28.47	0
PCA	42.94	0	34.55	0	37.98	0
GFK	52.56	0	26.15	0	39.21	0
TSL	47.22	0	36.10	0	42.97	1
JDA	57.30	0	**56.69**	**7**	44.34	1
OT-exact	49.96	0	50.47	0	36.69	0
OT-IT	59.20	0	54.89	0	42.30	0
OT-Lap	61.07	0	56.10	3	43.20	0
OT-LpLq	**64.11**	**1**	55.45	0	46.42	1
OT-GL	63.90	**1**	55.88	2	**47.70**	**9**

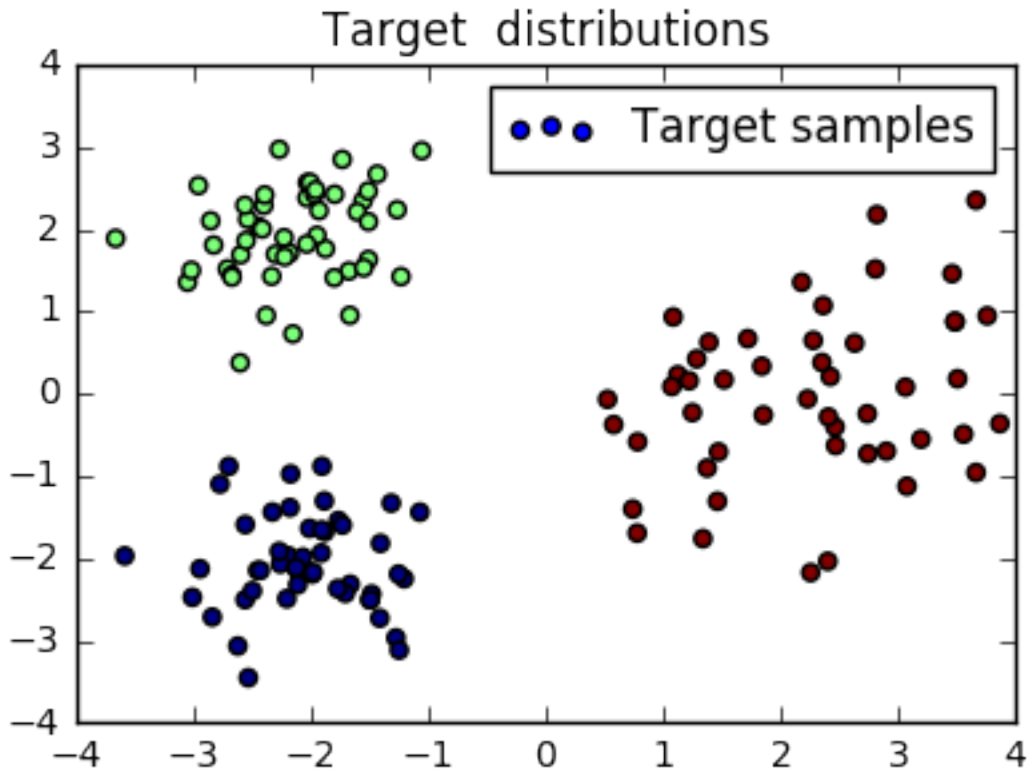
Discussion

- We report mean accuracy (ACC) and the number of time the method have been the best among all possible adaptation pairs.
- OT works very well on digits and object recognition (+7% and +3% wrt JDA).
- Good but not best on face recognition (-.5% wrt JDA).

In POT

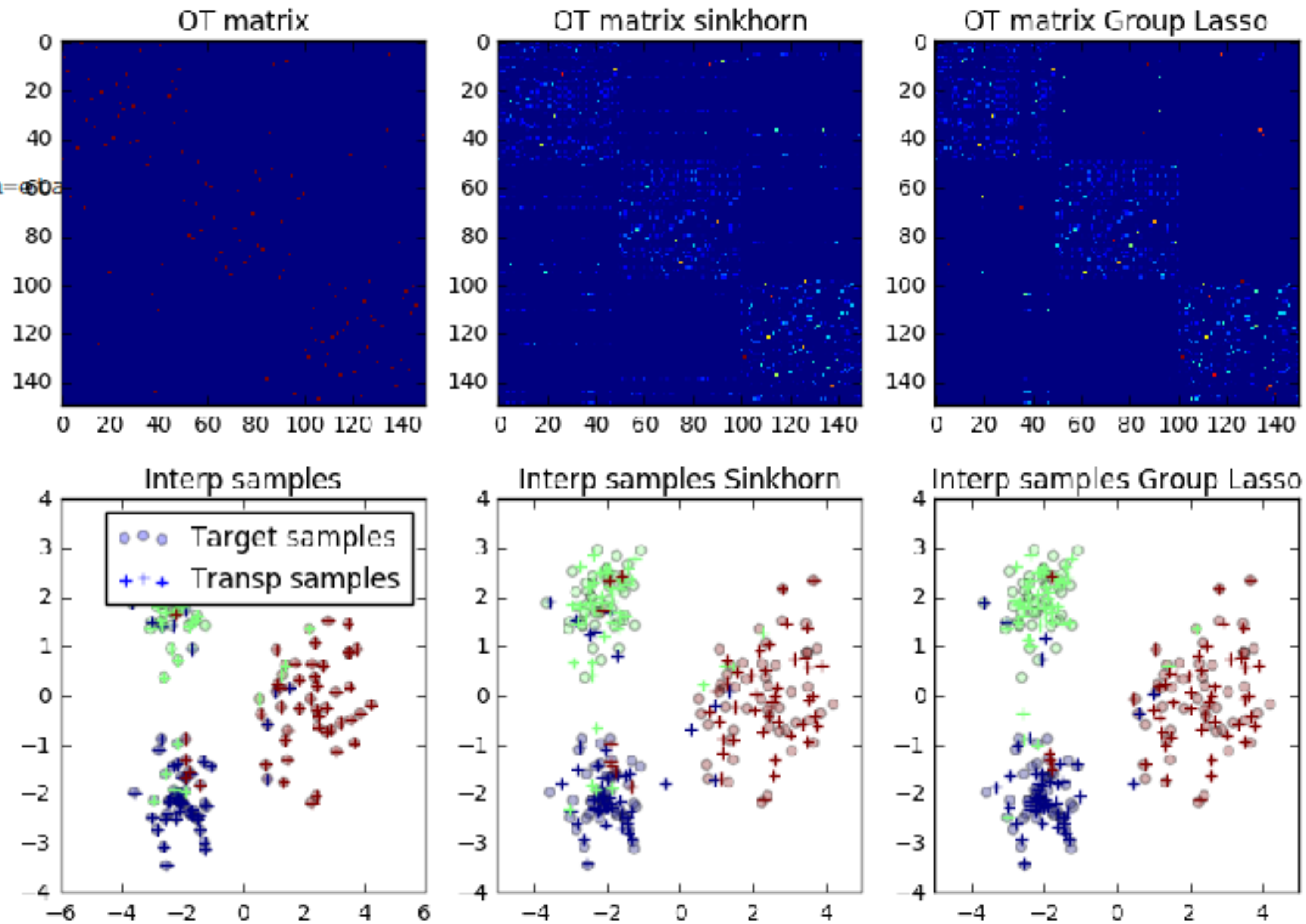
```
In [2]: n=150 # nb samples in source and target datasets

        xs,ys=ot.datasets.get_data_classif('3gauss',n)
        xt,yt=ot.datasets.get_data_classif('3gauss2',n)
```





In POT

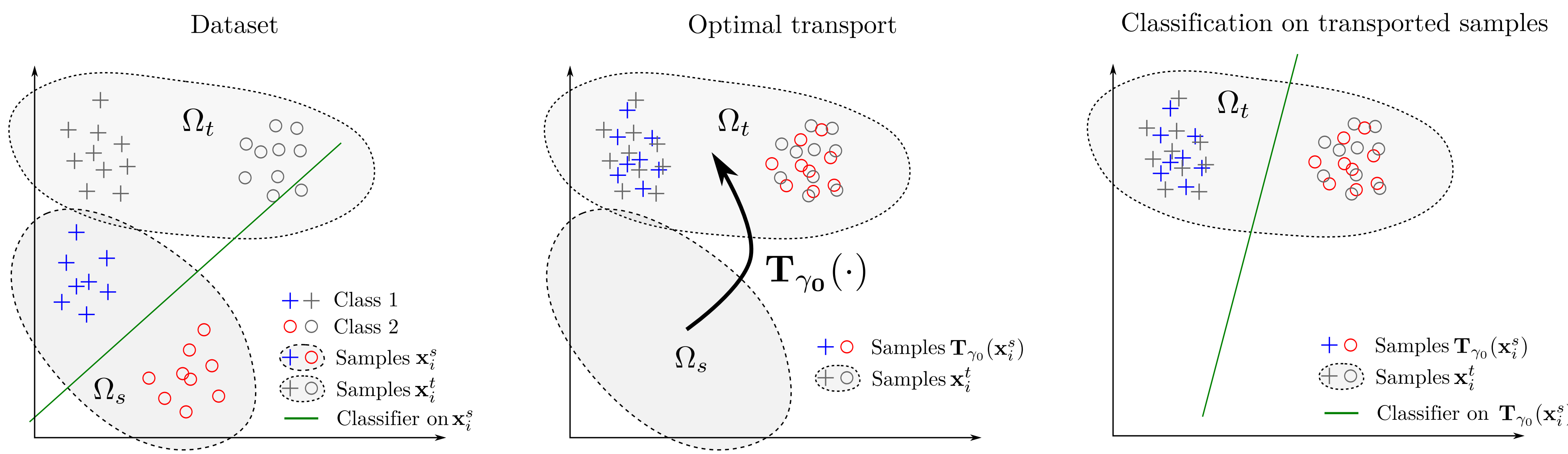
```
In [4]: # LP problem
        da_emd=ot.da.OTDA()        # init class
        da_emd.fit(xs,xt)          # fit distributions
        xst0=da_emd.interp()       # interpolation of source samples


        # sinkhorn regularization
        lambd=1e-1
        da_entrop=ot.da.OTDA_sinkhorn()
        da_entrop.fit(xs,xt,reg=lambd)
        xsts=da_entrop.interp()

        # Group lasso regularization
        reg=1e-1
        eta=1e0
        da_lpl1=ot.da.OTDA_lpl1()
        da_lpl1.fit(xs,ys,xt,reg=lambd,eta=eta)
        xstg=da_lpl1.interp()
```



Optimal Transport for domain adaptation

introduction to domain adaptation
regularization helps
out of samples formulation

Mapping estimation for discrete optimal transport



Why estimate the mapping?

- Out of sample problem.
- Solving optimization problem every time the dataset changes.
- Transporting a very large number of samples.
- Interpretability (depending on the mapping model).

How to estimate the mapping ?

- Go back to Monge formulation? No!
- Can use the barycentric mapping on the data samples.
- We want to fit the barycentric mapping but also introduce smoothness.

Mapping estimation

Problem formulation [Perrot et al., 2016]

$$\mathop{\arg\min}_{T\in\mathcal{H},\gamma\in\mathcal{P}} \quad f(\gamma,T) = \underbrace{\lambda_\gamma \left\langle \gamma, \mathbf{C}\right\rangle_{\mathcal{F}}}_{\text{OT loss}} + \underbrace{\|T(\mathbf{X}_s) - n_s\gamma\mathbf{X}_t\|_{\mathcal{F}}^2}_{\text{Mapping data fitting}} + \underbrace{\lambda_T R(T)}_{\text{Mapping reg.}} \tag{10}$$

where

- $\mathbf{X}_s = [\mathbf{x}_1^s, \dots, \mathbf{x}_{n_s}^s]^\top$ and $\mathbf{X}_t = [\mathbf{x}_1^t, \dots, \mathbf{x}_{n_t}^t]^\top$ are the source and target datasets,
- $T(\cdot)$ is applied for each elements of the above matrices,
- $n_s\gamma\mathbf{X}_t$ is the barycentric mapping for source samples with uniform weights,
- $\mathcal{H}$ is the space of transformations (more details later),
- $R(\cdot)$ is a regularization term controlling the complexity of T.

Convexity and optimization

- Problem (10) is jointly convex if $R(\cdot)$ is convex and $\mathcal{H}$ is a convex set.
- We propose to use a block coordinate descent to solve the problem.

Mapping estimation interpretation

Regression problem

$$\underset{T\in\mathcal{H},\gamma\in\mathcal{P}}{\arg\min} \quad f(\gamma,T) = \underbrace{\lambda_\gamma \left\langle \gamma, \mathbf{C} \right\rangle_{\mathcal{F}} + \|T(\mathbf{X}_s) - n_s\gamma\mathbf{X}_t\|_{\mathcal{F}}^2}_{\text{Data fitting}} + \underbrace{\lambda_T R(T)}_{\text{Regularization}}$$

- Mapping aim at fitting the barycentric mapping.
- Allow for a mapping model that can be reused (out of sample).
- Can we do OT then estimation [Perrot and Habrard, 2015]?

Regularized optimal transport

$$\underset{T\in\mathcal{H},\gamma\in\mathcal{P}}{\arg\min} \quad f(\gamma,T) = \underbrace{\lambda_\gamma \left\langle \gamma, \mathbf{C} \right\rangle_{\mathcal{F}}}_{\text{OT loss}} + \underbrace{\|T(\mathbf{X}_s) - n_s\gamma\mathbf{X}_t\|_{\mathcal{F}}^2 + \lambda_T R(T)}_{\text{OT regularization}}$$

- Adapt OT to the mapping .
- Model based regularization for OT.

Mapping family $\mathcal{H}$

Linear transformations

$$\mathcal{H} = \left\{ T : \forall \mathbf{x} \in \Omega, T(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \right\}. \tag{11}$$

- $\mathbf{L}$ is a $d \times d$ real matrix.
- $R(T) = \|\mathbf{L} - \mathbf{I}\|_{\mathcal{F}}^2$ where $\mathbf{I}$ is the identity matrix.
- Update is a classical linear least square regression.

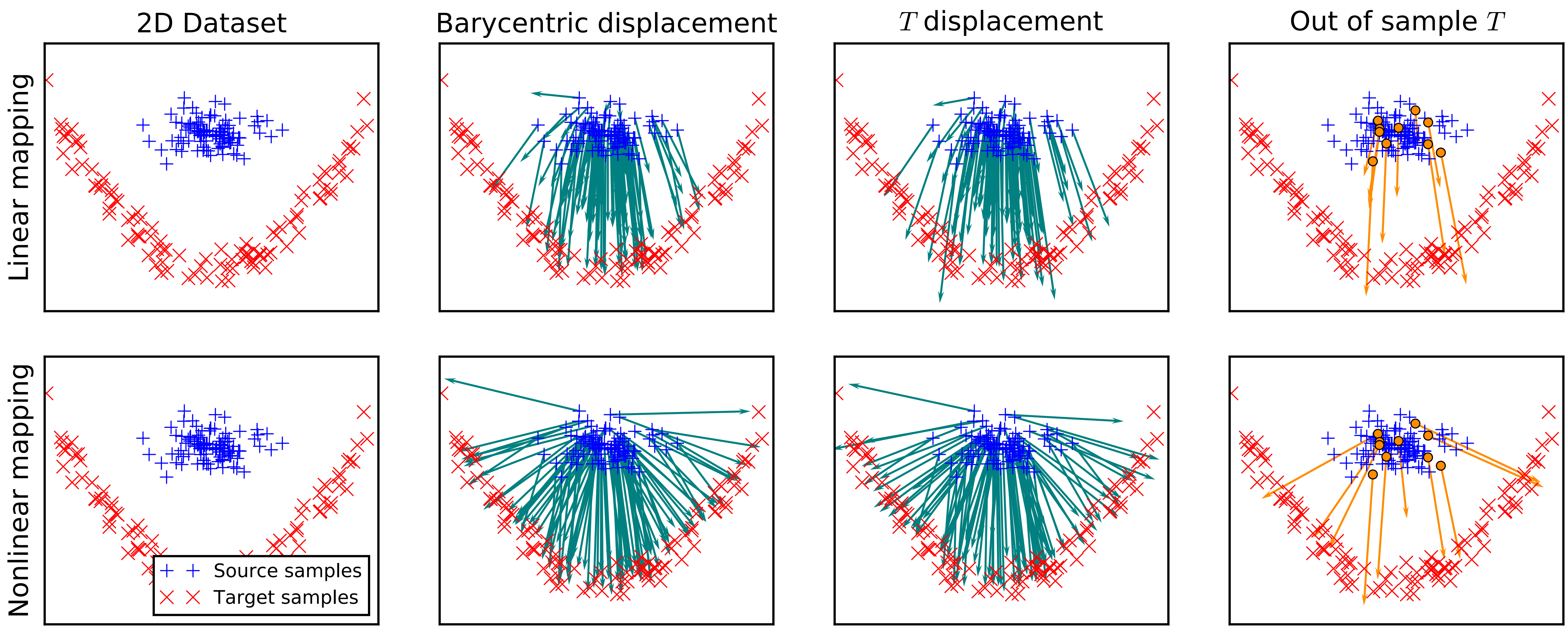
Nonlinear transformations

$$\mathcal{H} = \left\{ T : \forall \mathbf{x} \in \Omega, T(\mathbf{x}) = k_{\mathbf{X}_s}(\mathbf{x}^T)\mathbf{L} \right\} \tag{12}$$

- $k_{\mathbf{X}_s}(\mathbf{x}^T) = \begin{pmatrix} k(\mathbf{x}, \mathbf{x}_1^s) & k(\mathbf{x}, \mathbf{x}_2^s) & \cdots & k(\mathbf{x}, \mathbf{x}_{n_s}^s) \end{pmatrix}$.
- $k(\cdot, \cdot)$ is a positive definite kernel.
- $\mathbf{L}$ is a $n_s \times d$ real matrix.
- Update is a classical kernel least square regression.

For both models we can add a bias to get affine transformations.

Illustrative example



Clown 2D dataset

- Clearly a non-linear mapping.
- The mapping model can control the barycentric mapping.

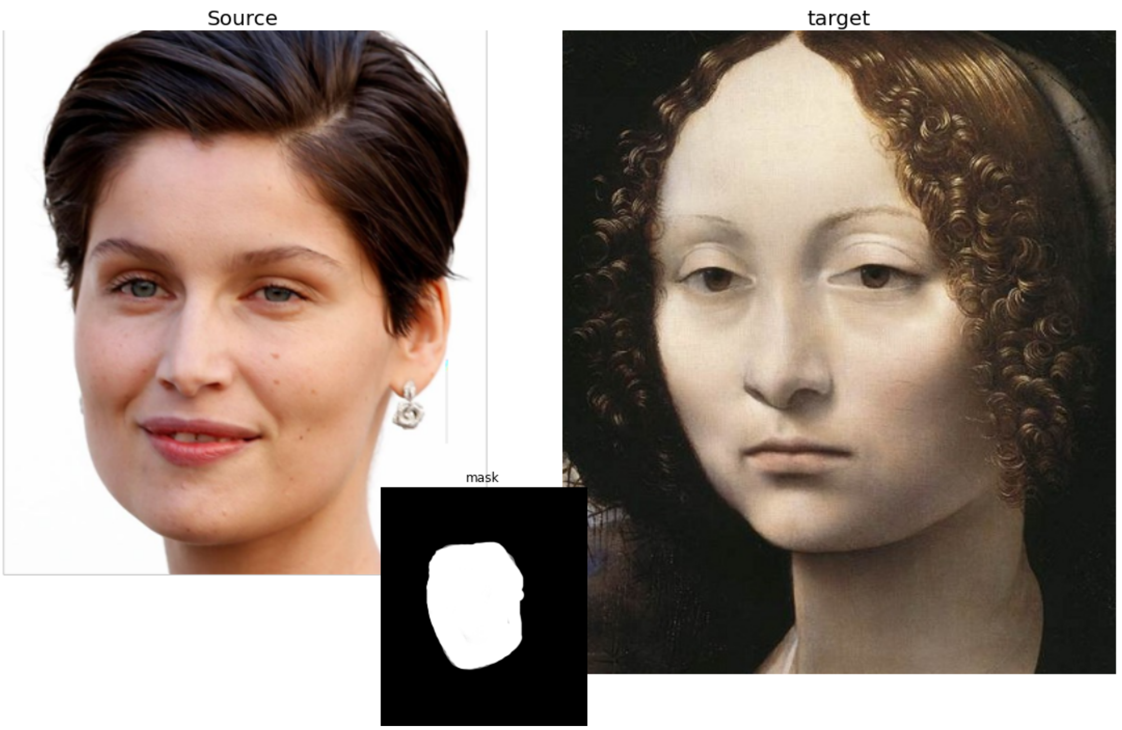
Domain adaptation: Caltech-Office dataset

Task	1NN	GFK	SA	OT	L1L2	OTE	OTLin		OTLinB		OTKer		OTKerB	
							T	γ	T	γ	T	γ	T	γ
$D \rightarrow W$	89.5	93.3	95.6	77.0	95.7	95.7	97.3	97.3	97.3	97.3	98.4	**98.5**	**98.5**	**98.5**
$D \rightarrow A$	62.5	77.2	88.5	70.8	74.9	74.8	85.7	85.7	85.8	85.8	**89.9**	**89.9**	89.5	89.5
$D \rightarrow C$	51.8	69.7	**79.0**	68.1	67.8	68.0	77.2	77.2	77.4	77.4	69.1	69.2	69.3	69.3
$W \rightarrow D$	99.2	**99.8**	99.6	74.1	94.4	94.4	99.4	99.4	**99.8**	**99.8**	97.2	97.2	96.9	96.9
$W \rightarrow A$	62.5	72.4	79.2	67.6	71.3	71.3	**81.5**	**81.5**	81.4	81.4	78.5	78.3	78.5	78.8
$W \rightarrow C$	59.5	63.7	55.0	63.1	67.8	67.8	**75.9**	**75.9**	75.4	75.4	72.7	72.7	65.1	63.3
$A \rightarrow D$	65.2	75.9	**83.8**	64.6	70.1	70.5	80.6	80.6	80.4	80.5	65.6	65.5	71.9	71.5
$A \rightarrow W$	56.8	68.0	74.6	66.8	67.2	67.3	**74.6**	**74.6**	74.4	74.4	66.4	64.8	70.0	68.9
$A \rightarrow C$	70.1	75.7	79.2	70.4	74.1	74.3	81.8	81.8	81.6	81.6	84.4	84.4	**84.5**	**84.5**
$C \rightarrow D$	75.9	79.5	85.0	66.0	69.8	70.2	87.1	87.1	**87.2**	**87.2**	70.1	70.0	78.6	78.6
$C \rightarrow W$	65.2	70.7	74.4	59.2	63.8	63.8	78.3	78.3	78.5	78.5	80.0	**80.4**	73.5	73.4
$C \rightarrow A$	85.8	87.1	89.3	75.2	76.6	76.7	**89.9**	**89.9**	89.7	89.7	82.4	82.2	83.6	83.5
Mean	70.3	77.8	81.9	68.6	74.5	74.6	**84.1**	**84.1**	**84.1**	**84.1**	79.6	79.4	80.0	79.7

Discussion

- Visual adaptation on DA deep learning features (decaf6 [Donahue et al., 2014])
- Parameter validation performed using circular validation.
- Clear advantage to the mapping estimation methods.

Seamless copy in images



Poisson image editing [Pérez et al., 2003]

- Let f_t be the target image and f_s the source image and a region of the image Ω.
- Poisson editing aim at solving f with Dirichlet boundary conditions

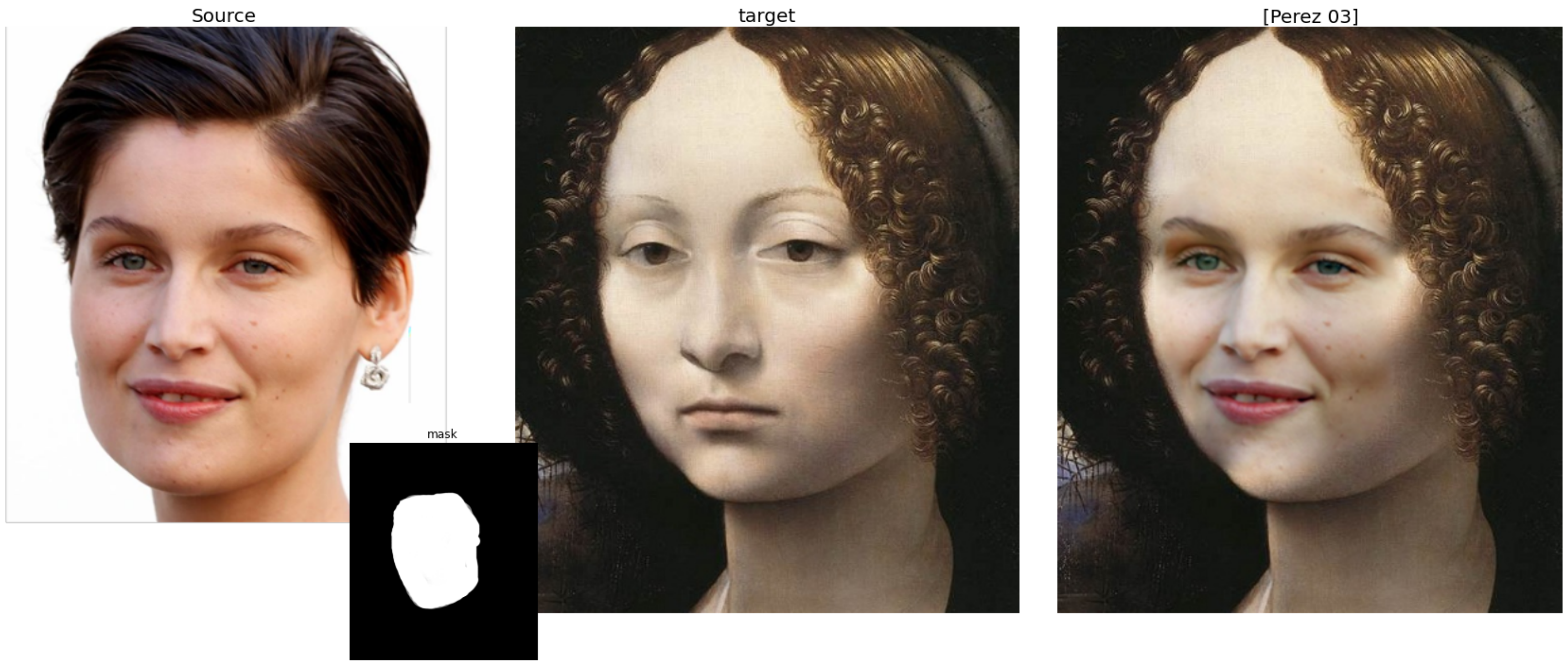
$$\min_f \int\int_\Omega |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{13}$$

- Here $\mathbf{v} = \nabla f_s|_\Omega$ is given as the gradient from the source image f_s over Ω.
- Equivalent so solving the following Poisson equation [Pérez et al., 2003]

$$\Delta f = \text{div } \mathbf{v} \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{14}$$

- Using first order discretization, the problem is a large sparse linear system.

Seamless copy in images



Poisson image editing [Pérez et al., 2003]

- Let f_t be the target image and f_s the source image and a region of the image Ω.
- Poisson editing aim at solving f with Dirichlet boundary conditions

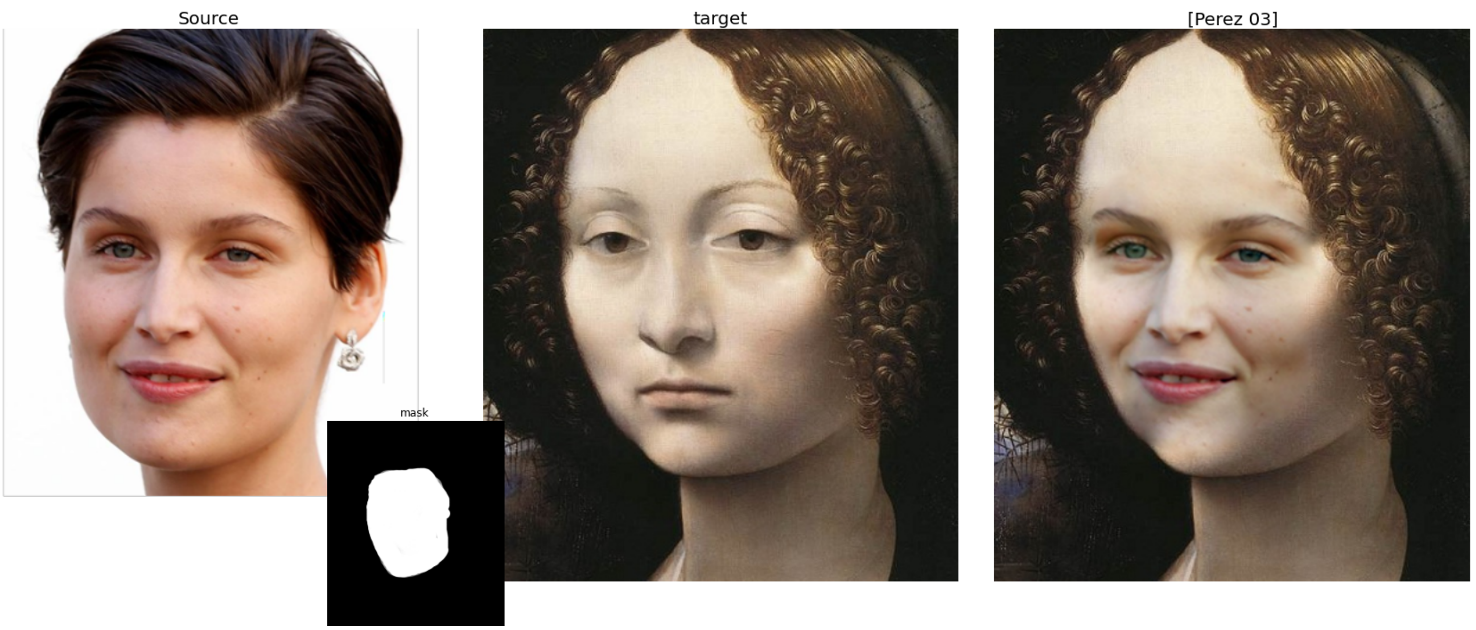
$$\min_f \int\int_\Omega |\nabla f - \mathbf{v}|^2 \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{13}$$

- Here $\mathbf{v} = \nabla f_s|_\Omega$ is given as the gradient from the source image f_s over Ω.
- Equivalent so solving the following Poisson equation [Pérez et al., 2003]

$$\Delta f = \text{div } \mathbf{v} \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{14}$$

- Using first order discretization, the problem is a large sparse linear system.

Seamless copy with gradient adaptation



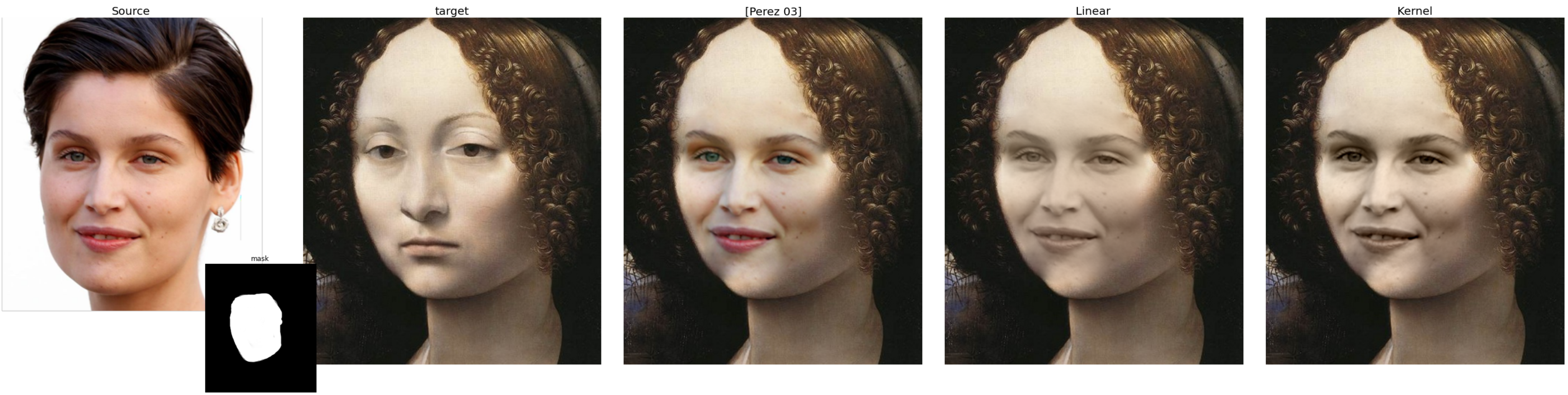
Poisson image editing with gradient adaptation

- Poisson image editing leads to false colors in practice.
- We propose to adapt the gradients from the source to the target domain:

$$\Delta f = \text{div } T_{s\to t}(\mathbf{v}) \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{15}$$

- $T_{s\to t} : \mathbb{R}^6 \to \mathbb{R}^6$ is the mapping between gradients of the source and target images in the domain.

Seamless copy with gradient adaptation



Poisson image editing with gradient adaptation

- Poisson image editing leads to false colors in practice.
- We propose to adapt the gradients from the source to the target domain:

$$\Delta f = \text{div}\, T_{s\to t}(\mathbf{v}) \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{15}$$

- $T_{s\to t} : \mathbb{R}^6 \to \mathbb{R}^6$ is the mapping between gradients of the source and target images in the domain.

Seamless copy with gradient adaptation

Poisson image editing with gradient adaptation

- Poisson image editing leads to false colors in practice.
- We propose to adapt the gradients from the source to the target domain:

$$\Delta f = \text{div } T_{s \to t}(\mathbf{v}) \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{15}$$

- $T_{s \to t} : \mathbb{R}^6 \to \mathbb{R}^6$ is the mapping between gradients of the source and target images in the domain.

Seamless copy with gradient adaptation

Poisson image editing with gradient adaptation

- Poisson image editing leads to false colors in practice.
- We propose to adapt the gradients from the source to the target domain:

$$\Delta f = \text{div}\, T_{s \to t}(\mathbf{v}) \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{15}$$

- $T_{s \to t} : \mathbb{R}^6 \to \mathbb{R}^6$ is the mapping between gradients of the source and target images in the domain.

Seamless copy with gradient adaptation

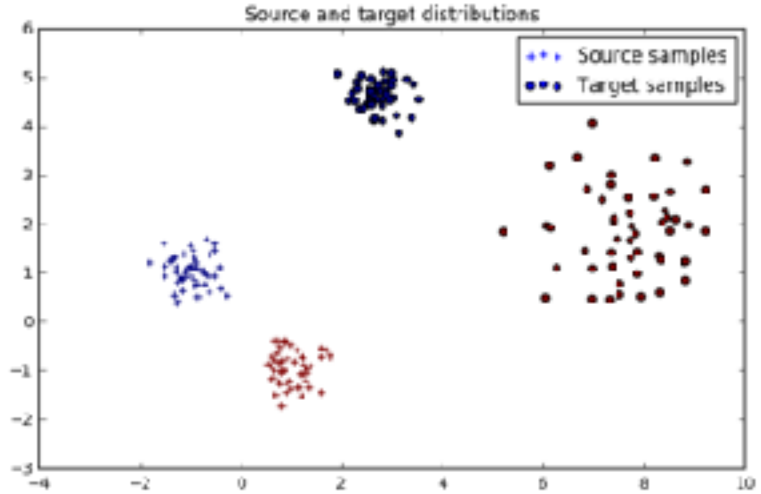
Poisson image editing with gradient adaptation

- Poisson image editing leads to false colors in practice.
- We propose to adapt the gradients from the source to the target domain:

$$\Delta f = \text{div}\, T_{s\to t}(\mathbf{v}) \quad \text{over } \Omega, \quad \text{with} \quad f|_{\partial\Omega} = f_t|_{\partial\Omega}. \tag{15}$$

- $T_{s\to t} : \mathbb{R}^6 \to \mathbb{R}^6$ is the mapping between gradients of the source and target images in the domain.

In POT

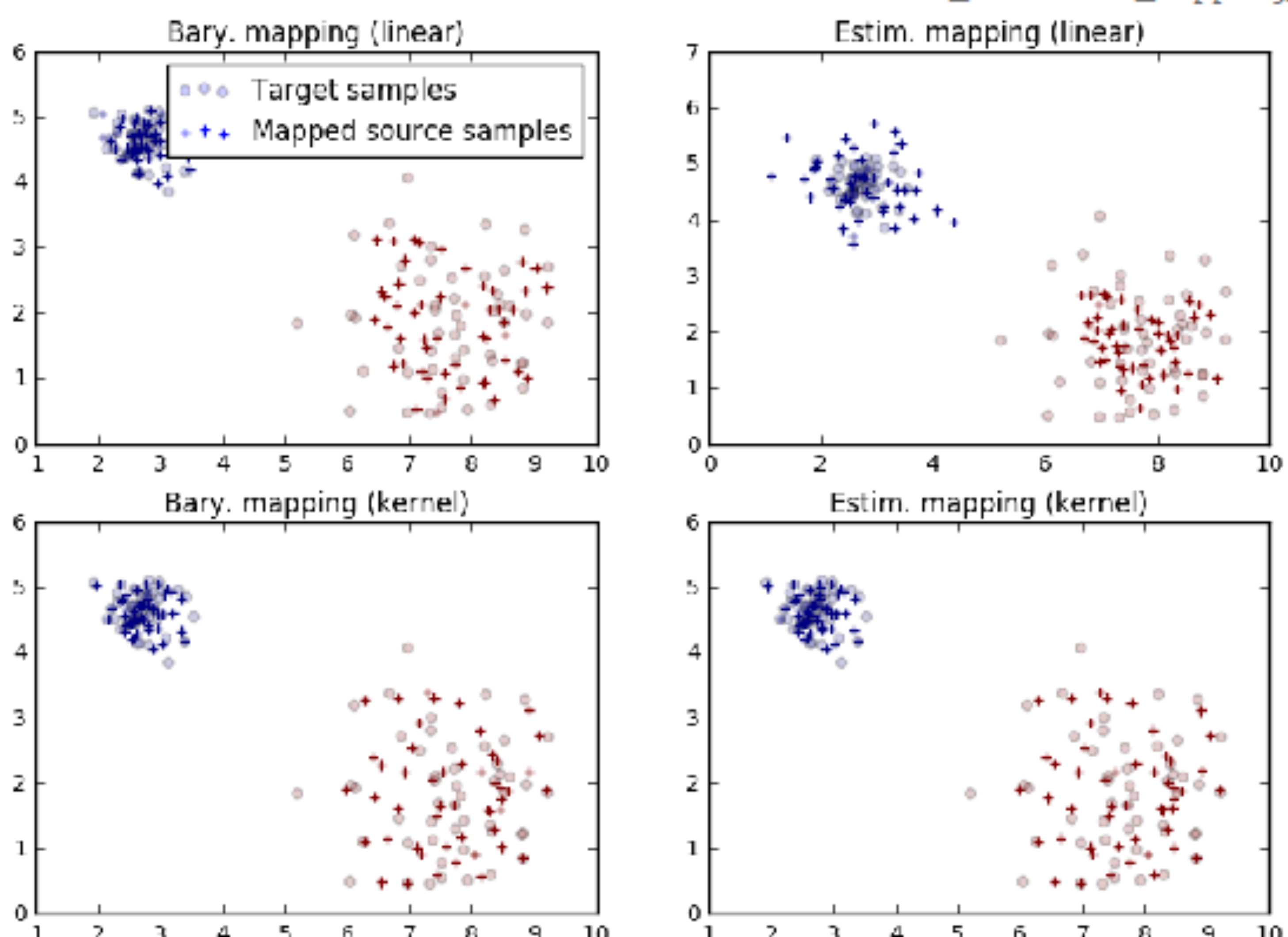


0.1.3 OT linear mapping estimation

```
In [4]: eta=1e-8    # quadratic regularization for regression
        mu=1e0      # weight of the OT linear term
        bias=True   # estimate a bias

        ot_mapping=ot.da.OTDA_mapping_linear()
        ot_mapping.fit(xs,xt,mu=mu,eta=eta,bias=bias,numItermax = 2

        xst=ot_mapping.predict(xs) # use the estimated mapping
        xst0=ot_mapping.interp()    # use barycentric mapping
```

0.1.4 OT kernel mapping estimation

```
In [5]: eta=1e-5    # quadratic regularization for regression
        mu=1e-1      # weight of the OT linear term
        bias=True   # estimate a bias
        sigma=1     # sigma bandwidth fot gaussian kernel


        ot_mapping_kernel=ot.da.OTDA_mapping_kernel()
        ot_mapping_kernel.fit(xs,xt,mu=mu,eta=eta,sigma=sigma

        xst_kernel=ot_mapping_kernel.predict(xs) # use the es
        xst0_kernel=ot_mapping_kernel.interp()   # use baryce
```

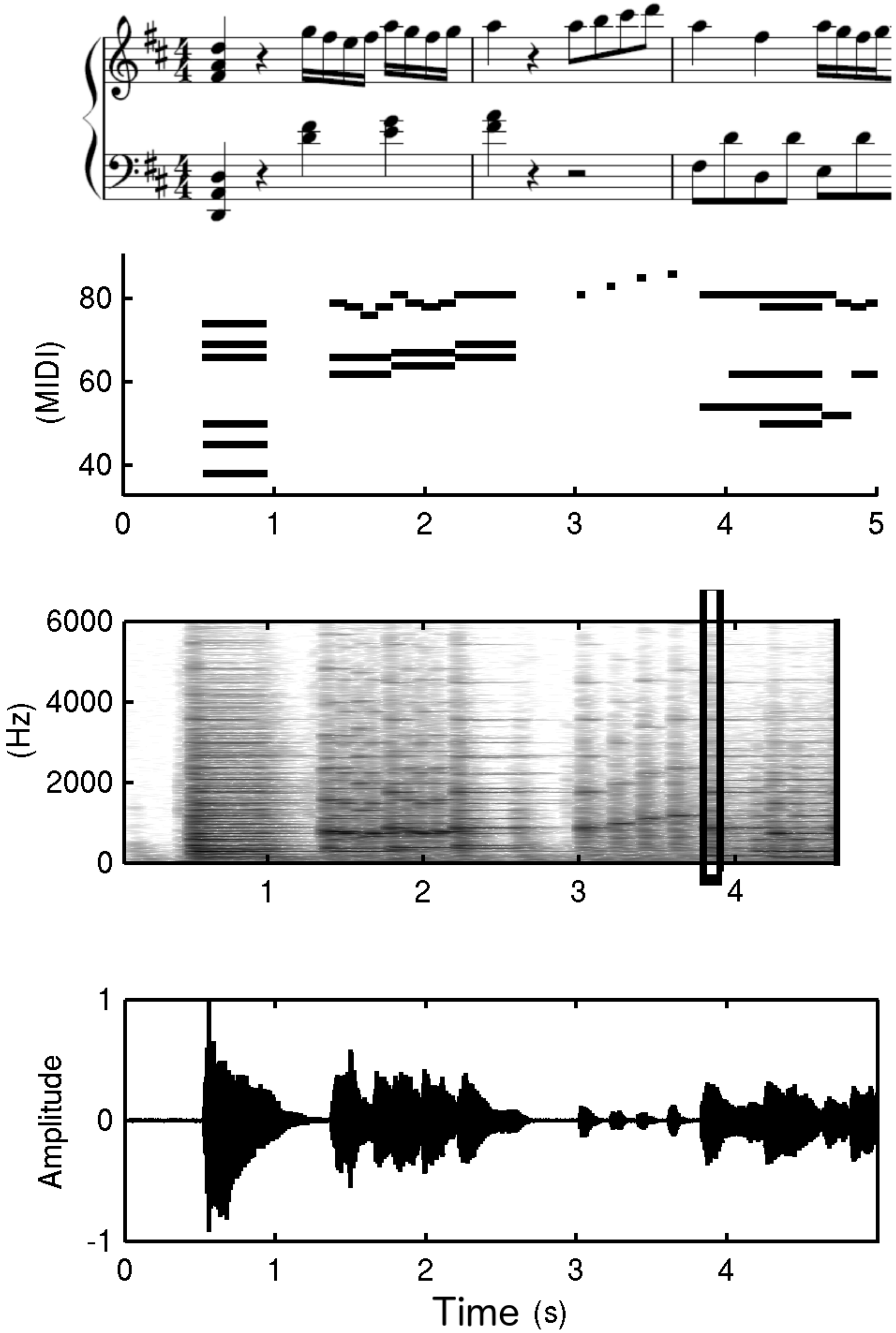


Optimal Transport for music transcription

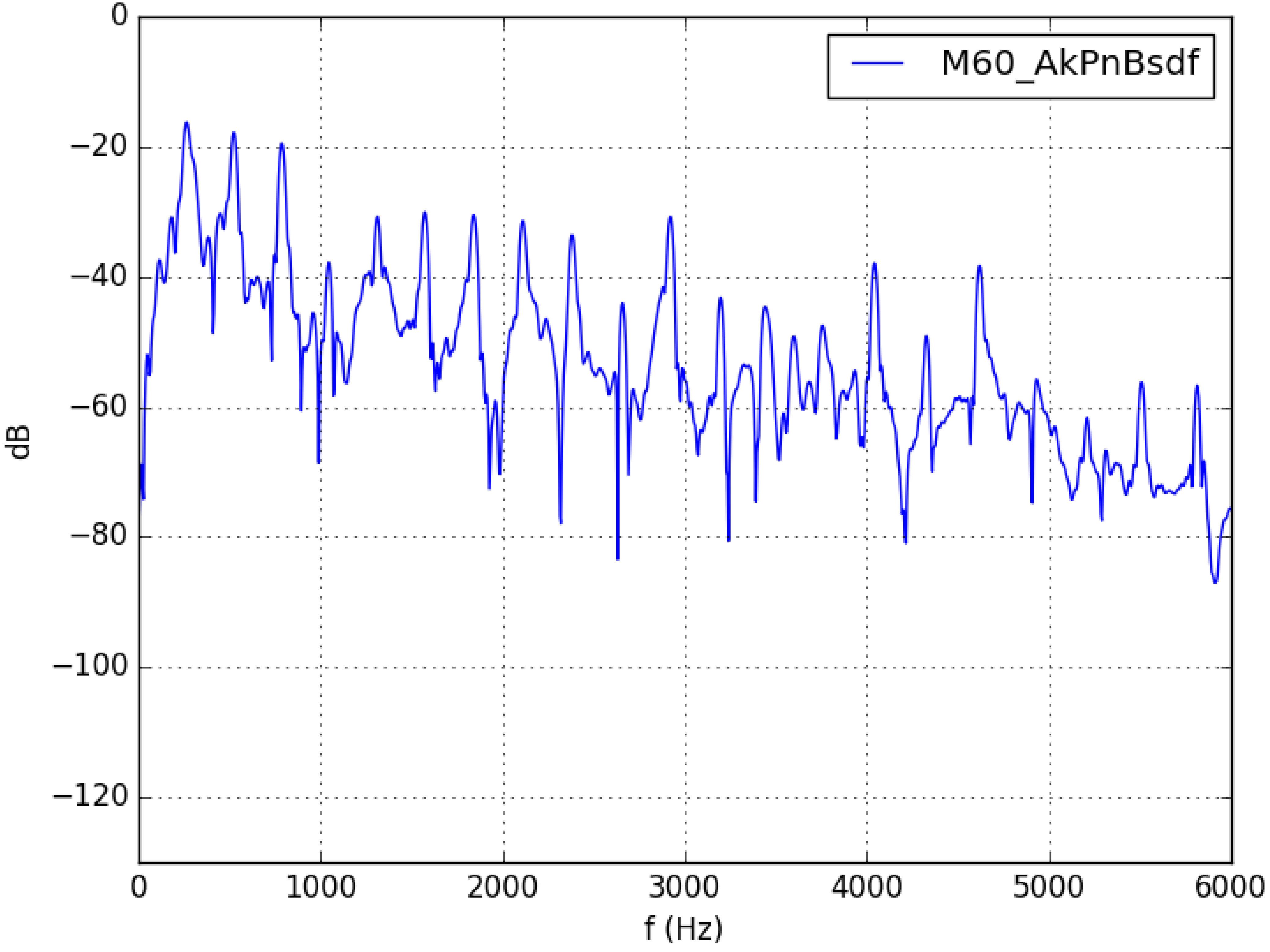
introduction to problem
a solution with OT
some results

Joint work with Rémi Flamary, Cédric Févotte, Valentin Emiya

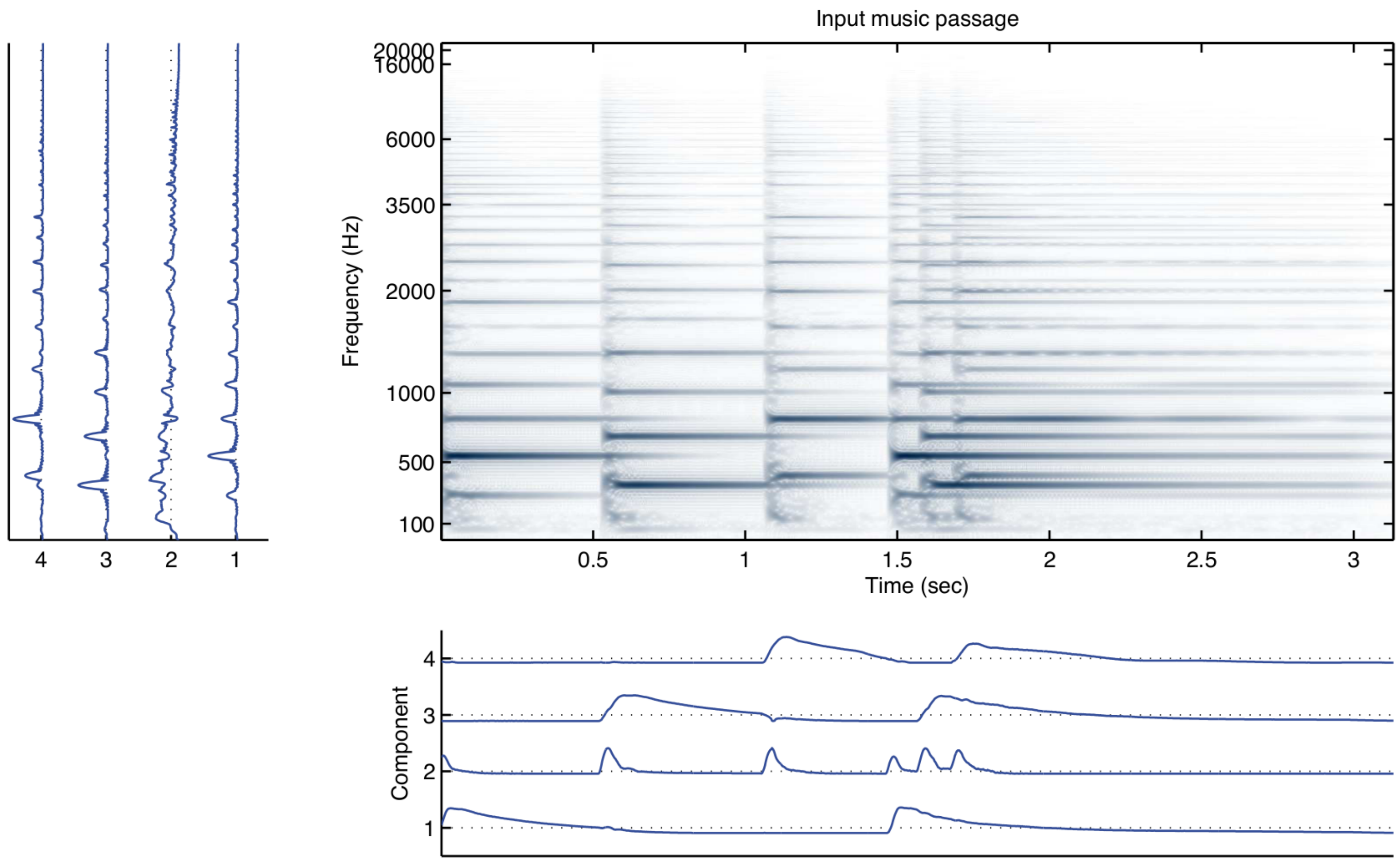
Automatic music transcription : tracking note spectra



Short-term spectrum of notes



Baseline: PLCA (Smaragdis et al., 2006)



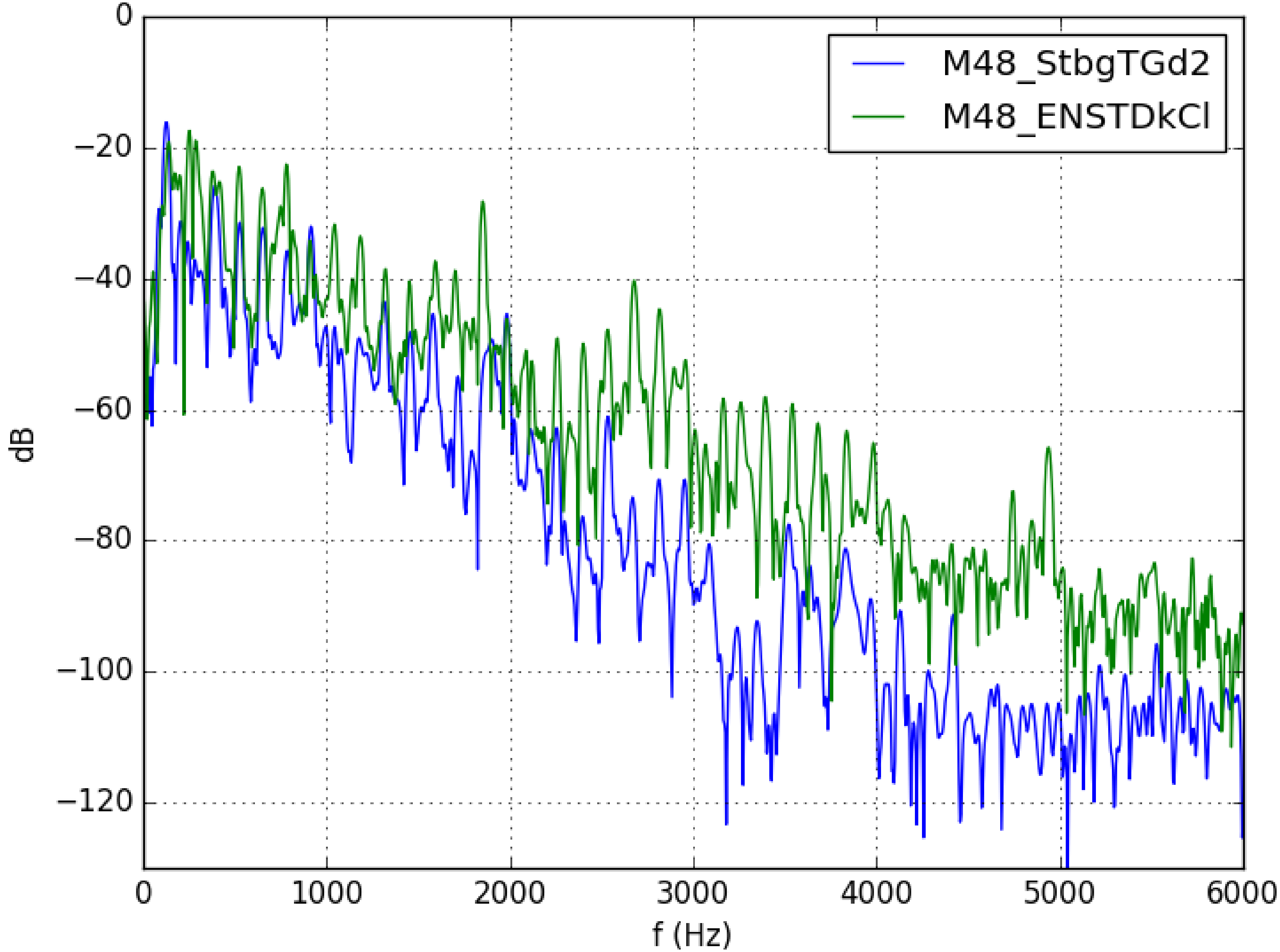
(from Smaragdis 2013)

Estimate transcription $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_N] \in \mathbb{R}_+^{K \times N}$ from $\mathbf{V} \in \mathbb{R}_+^{M \times N}$ and $\mathbf{W} \in \mathbb{R}_+^{M \times K}$ by solving

$$\min_{\mathbf{H} \geq 0} D_{\mathrm{KL}}\left(\mathbf{V} \,|\, \mathbf{WH}\right) \;\; s.t. \;\; \forall n, \|\mathbf{h}_n\|_1 = 1$$

where $D_{\mathrm{KL}}\left(\mathbf{v} \,|\, \widehat{\mathbf{v}}\right) = \sum_i v_i \log\left(v_i / \widehat{v}_i\right)$ and $D_{\mathrm{KL}}\left(\mathbf{V} \,\middle|\, \widehat{\mathbf{V}}\right) = \sum_n D_{\mathrm{KL}}\left(\mathbf{v}_n \,|\, \widehat{\mathbf{v}}_n\right)$

Comparing two note spectra

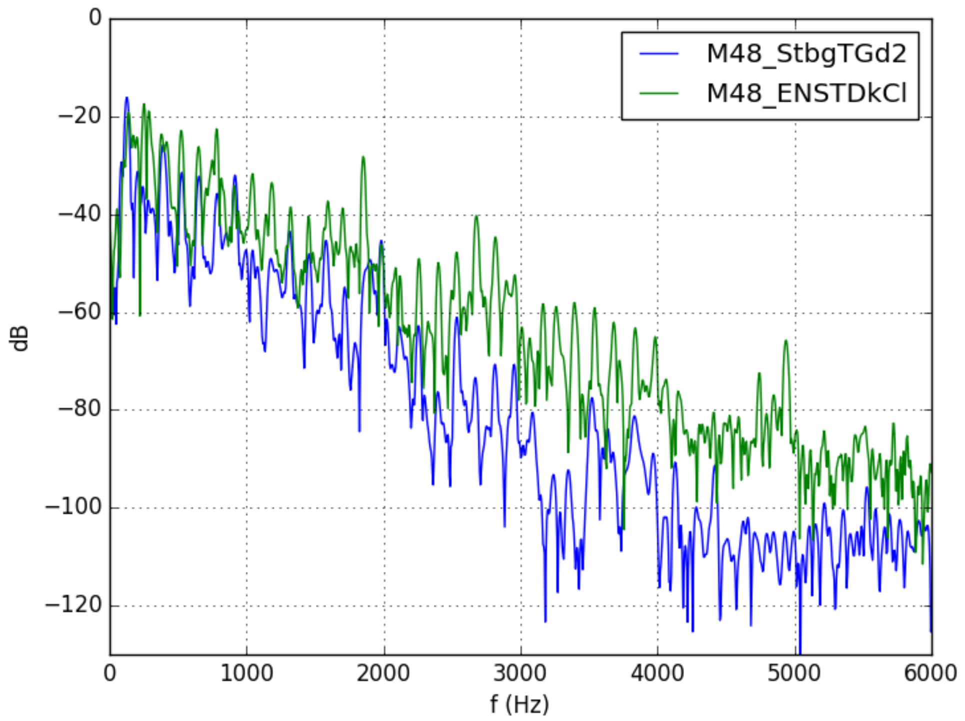


Comparing note spectra with usual metrics

Usual metrics (Euclidean, KL, IS) are separable:

$$d_p(\mathbf{u}, \mathbf{v}) = \sum_i |u_i - v_i|^p$$

$$d_{\mathrm{KL}}(\mathbf{u}, \mathbf{v}) = \sum_i u_i \log(u_i / v_i)$$



Separability is good for designing solvers like PLCA, but...

Actual comparison: frequency-wise, variability in amplitudes.

Any variability in frequency is measured frequency-wise as a variability in amplitude. Some partials of a true note may be missed

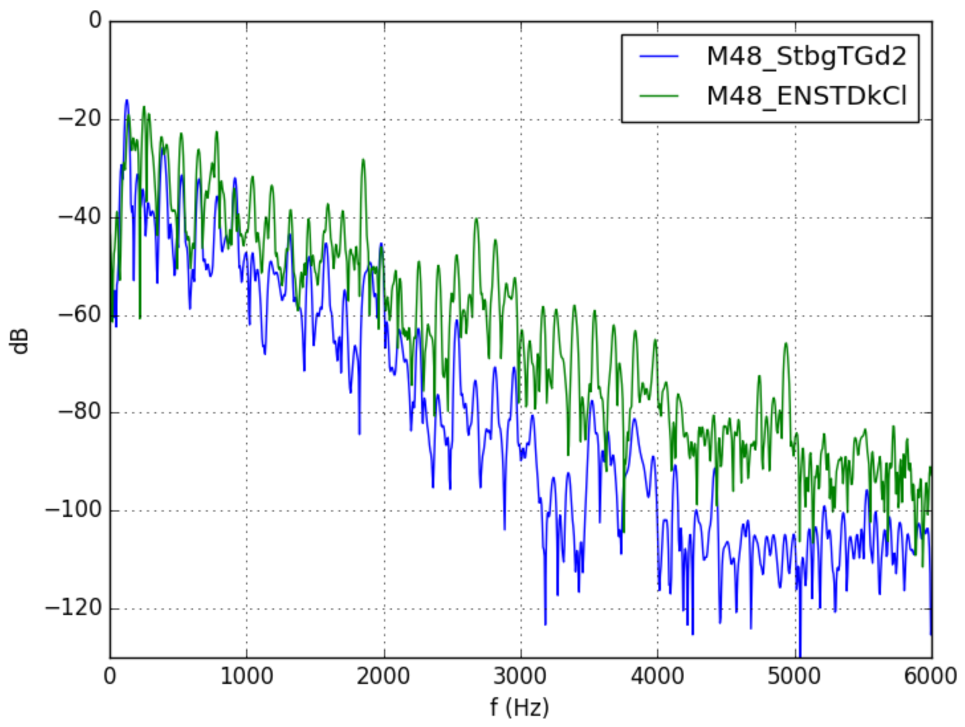
- the true note may not be well estimated
- other notes may be estimated: octave, fifth, and so on

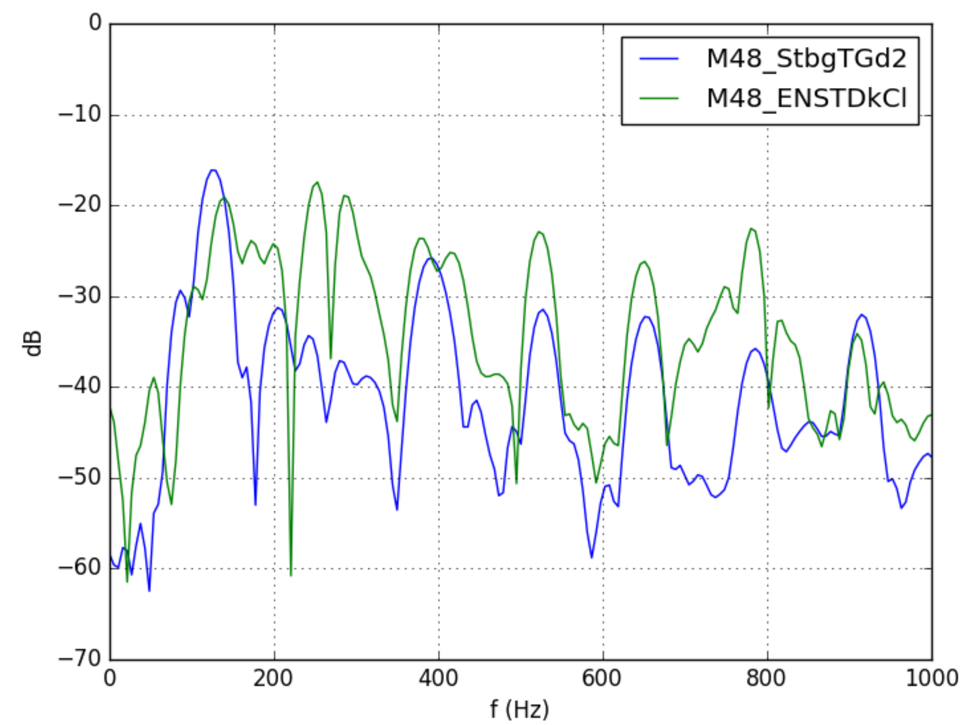
Variability in frequency and amplitude

- Variability in f0 due to tuning
- Variability in peak shape due window choice
- Variability in peak shape due to modulations
  - f0 modulation: varying pitch
  - beats due to multiple string
  - notes at unisson from various players
- Variability in frequency distribution due to inharmonicity

$$f_h = hf_0\sqrt{1+\beta h^2}$$

- Variability in amplitudes due to timber
- Variability in amplitudes in time due to attenuation and beats





(zoom)

Optimal Transport for music transcription

introduction to problem
a solution with OT
some results

Objective: finding the optimal transport from $\mathbf{u}$ to $\mathbf{v}$

Let us consider two vectors $\mathbf{u}$ and $\mathbf{v}$ to be compared by OT (e.g., two magnitude spectra). What is the best way to transport energy from $\mathbf{u}$ to $\mathbf{v}$?
Main issues:

1. how to transport energy from $\mathbf{u}$ to $\mathbf{v}$?

   $\rightarrow$ *using a transportation matrix* $\mathbf{T}$*.*

2. what does it cost?

   $\rightarrow$ *specify a (unitary-)cost matric* $\mathbf{C}$*.*

3. how to find the optimal transportation

   $\rightarrow$ *by solving a linear program.*

Transportation matrices $\mathbf{T}$

Let $\mathbf{u} \in \mathbb{R}_+^{N_u}$ and $\mathbf{v} \in \mathbb{R}_+^{N_v}$ such that $\|\mathbf{u}\|_1 = \|\mathbf{v}\|_1 = 1$.
We want to transport $\mathbf{u}$ to $\mathbf{v}$.
Let t_{ij} the part of u_i transported to v_j:

$$
\begin{array}{ccccc}
\mathbf{T} & j & & \mathbf{u} & \\
i & \begin{bmatrix} & & 0 & & & \\ 0 & 0.1 & 0.2 & 0 & 0.1 & 0.1 \\ & & 0.3 & & & \\ & & 0.1 & & & \end{bmatrix} & & \begin{bmatrix} \\ 0.5 \\ \\ \\ \end{bmatrix} & \begin{array}{c} \\ u_i \\ \\ \\ \end{array} \\
\mathbf{v} & \begin{bmatrix} & & 0.6 & & & \end{bmatrix} & & & \\
 & v_j & & &
\end{array}
$$

Transportation from $\mathbf{u}$ to $\mathbf{v}$ is valid iff

- For any i, u_i is distributed among all v_j's: $\sum_j t_{ij} = u_i$, i.e., $\mathbf{T}\mathbf{1}_{N_v} = \mathbf{u}$.
- For any j, all contributions to v_j sum up to v_j: $\sum_i t_{ij} = v_j$, i.e., $\mathbf{T}^T\mathbf{1}_{N_u} = \mathbf{v}$.

Transportation matrices $\mathbf{T}$

Let $\mathbf{u} \in \mathbb{R}_+^{N_u}$ and $\mathbf{v} \in \mathbb{R}_+^{N_v}$ such that $\|\mathbf{u}\|_1 = \|\mathbf{v}\|_1 = 1$.
We want to transport $\mathbf{u}$ to $\mathbf{v}$.
Let t_{ij} the part of u_i transported to v_j:

$$
\begin{array}{cc}
\mathbf{T} & \begin{array}{cccccc} & & j & & & \end{array} \\
\begin{array}{c} \\ i \\ \\ \\ \end{array} & \begin{bmatrix} & & 0 & & & \\ 0 & 0.1 & 0.2 & 0 & 0.1 & 0.1 \\ & & 0.3 & & & \\ & & 0.1 & & & \end{bmatrix} \\
\mathbf{v} & \begin{bmatrix} & & 0.6 & & & \end{bmatrix} \\
 & \begin{array}{cccccc} & & v_j & & & \end{array}
\end{array}
\qquad
\begin{array}{cc}
\mathbf{u} & \\
\begin{bmatrix} \\ 0.5 \\ \\ \\ \end{bmatrix} & \begin{array}{c} \\ u_i \\ \\ \\ \end{array}
\end{array}
$$

Definition: set of transportation matrices for $(\mathbf{u}, \mathbf{v})$

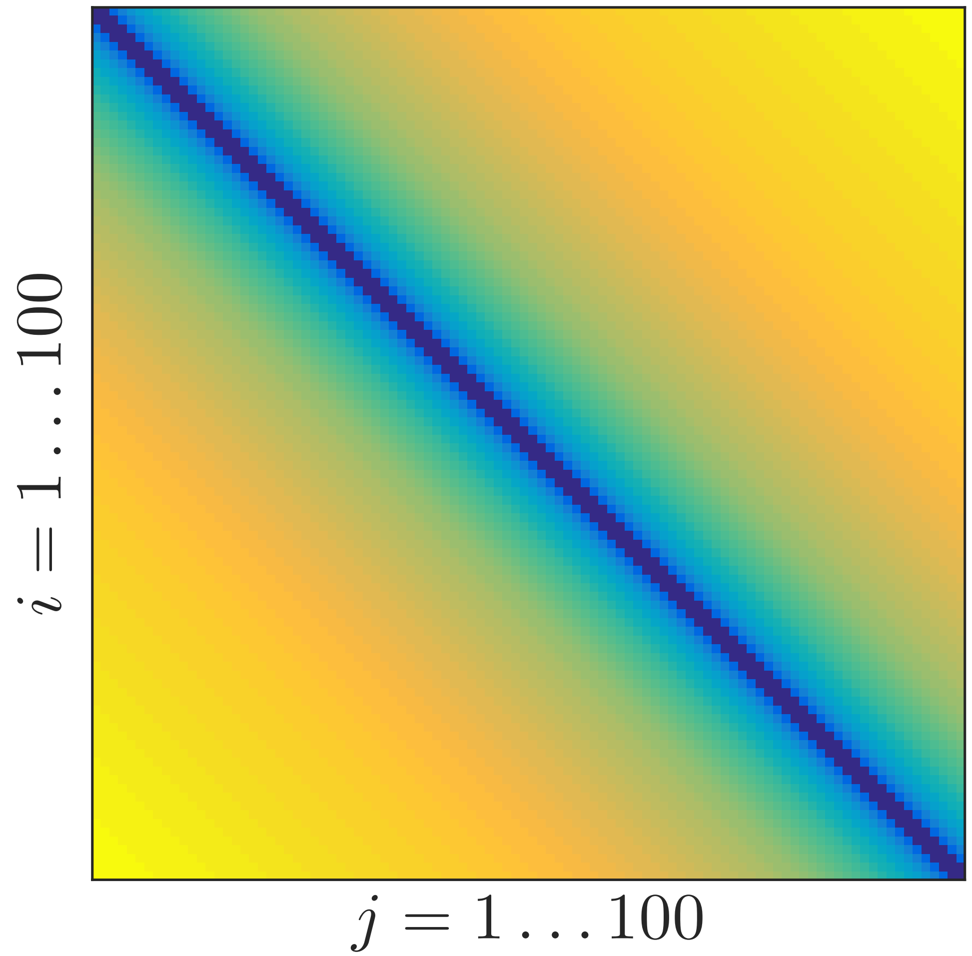
$$\Theta \triangleq \left\{ \mathbf{T} \in \mathbb{R}_+^{N_u \times N_v} : \mathbf{T}\mathbf{1}_{N_v} = \mathbf{u} \text{ and } \mathbf{T}^T \mathbf{1}_{N_u} = \mathbf{v} \right\}$$

Cost matrices **C**

Let $c_{ij} \geq 0$ be the cost to transport one unit from u_i to v_j: one may choose all c_{ij}'s and gather them into a matrix $\mathbf{C} \in \mathbb{R}_+^{N_u \times N_v}$.

Examples to compare two spectra:

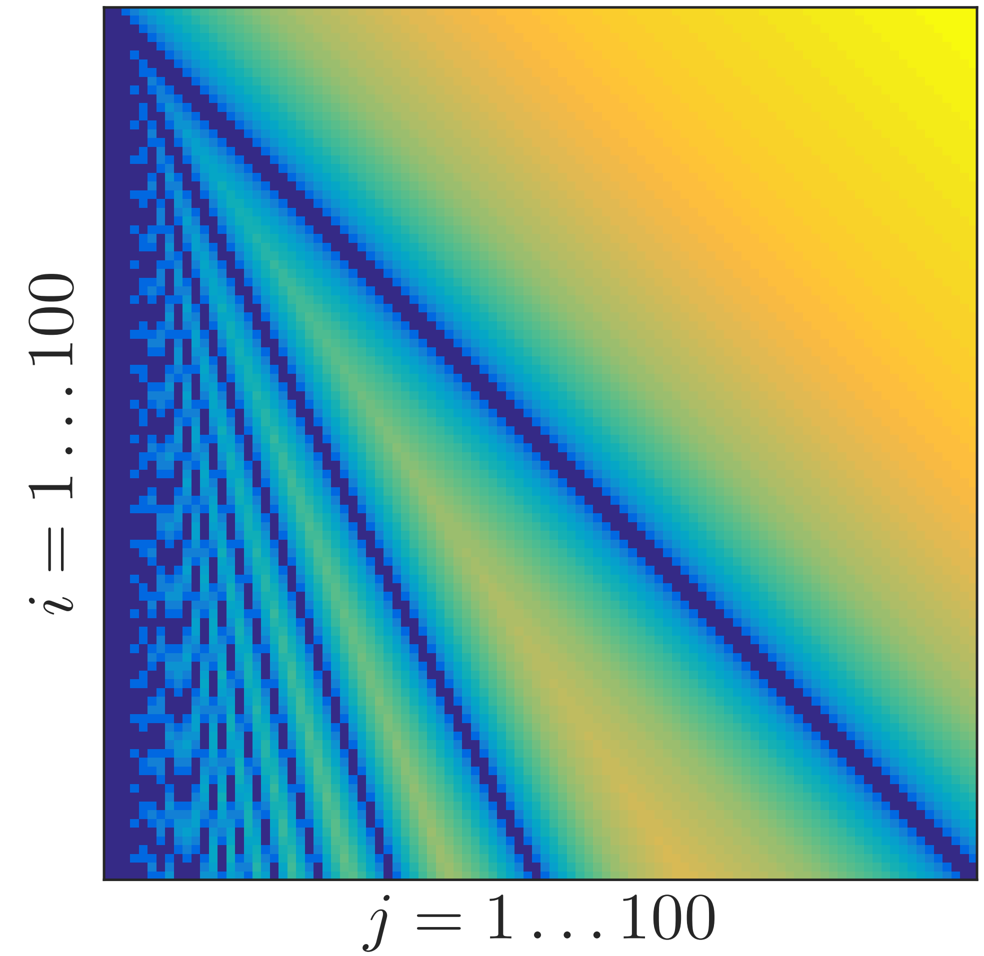
Quadratic cost $\mathbf{C}_2$ (log scale)



$c_{ij} = |f_i - f_j|^p$ ($p > 0$)

Only allows local displacements

Harmonic cost $\mathbf{C}_h$ (log scale)



Allows displacement of observed energy to any possible f0 candidate

$\rightarrow$ Transporting t_{ij} from u_i to v_j costs $c_{ij} t_{ij}$

Optimal transportation divergence as a optimization problem

Given a cost matrix $\mathbf{C}$, how to find the optimal transportation from $\mathbf{u}$ to $\mathbf{v}$?
$\rightarrow$ Find $\mathbf{T} \in \Theta$ such that the total cost $\sum_{ij} c_{ij} t_{ij}$ is minimal.

Optimal transportation divergence

$$D_{\mathbf{C}}(\mathbf{u} \,|\, \mathbf{v}) \triangleq \min_{T \geq 0} \langle \mathbf{T}, \mathbf{C} \rangle \;\; s.t. \;\; \mathbf{T}\mathbf{1}_{N_v} = \mathbf{u} \text{ and } \mathbf{T}^T \mathbf{1}_{N_u} = \mathbf{v}$$

where $\langle \mathbf{T}, \mathbf{C} \rangle = \sum_{ij} c_{ij} t_{ij}$.

- This is a linear program with convex constraints.
- Computing $D_{\mathbf{C}}(\mathbf{u} \,|\, \mathbf{v})$ implies solving an optimization problem
- Particular case $c_{ij} = |f_i - f_j|^p$: $D_{\mathbf{C}}(\mathbf{u} \,|\, \mathbf{v})$ is a metric called Wasserstein distance or earth mover's distance.
- In the general case, $D_{\mathbf{C}}(\mathbf{u} \,|\, \mathbf{v})$ is not a metric, we call it a divergence.

From PLCA to optimal spectral transportation with a fixed dictionary $\mathbf{W}$

PLCA

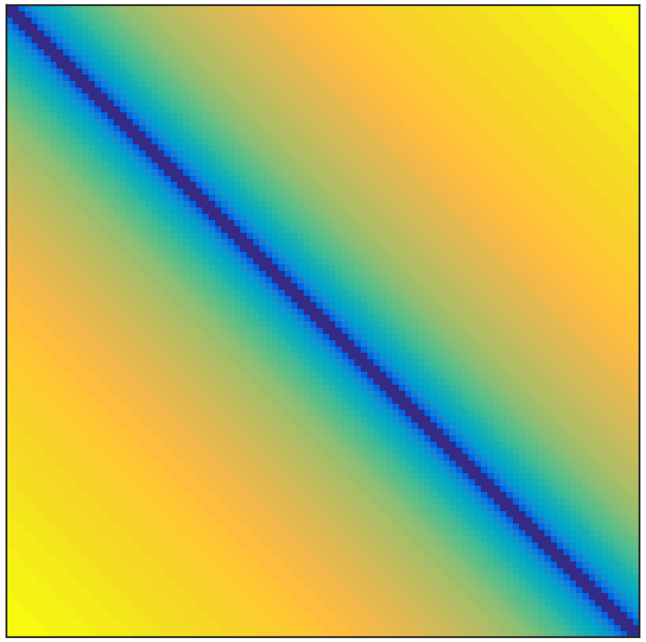
$$\min_{\mathbf{H} \geq 0} D_{\mathrm{KL}}(\mathbf{V}\,|\mathbf{WH}) \;\; s.t. \;\; \forall n, \|\mathbf{h}_n\|_1 = 1$$

Unmixing with OT

$$\min_{\mathbf{H} \geq 0} D_{\mathbf{C}}(\mathbf{V}\,|\mathbf{WH}) \;\; s.t. \;\; \forall n, \|\mathbf{h}_n\|_1 = 1$$

- **C** may be adjusted to allow local displacement (e.g., $c_{ij} = (f_i - f_j)^2$)
- Requires that columns of **W** to be appropriate note templates.
- Not robust to variability in spectral envelopes.

Quadratic cost $\mathbf{C}_2$ (log scale)



Harmonic-invariant transportation with a diract dictionary

Principle: allow energy at f_i to be transported to fundamental frequency $f_j = \frac{f_i}{q}$ with any positive integer q .

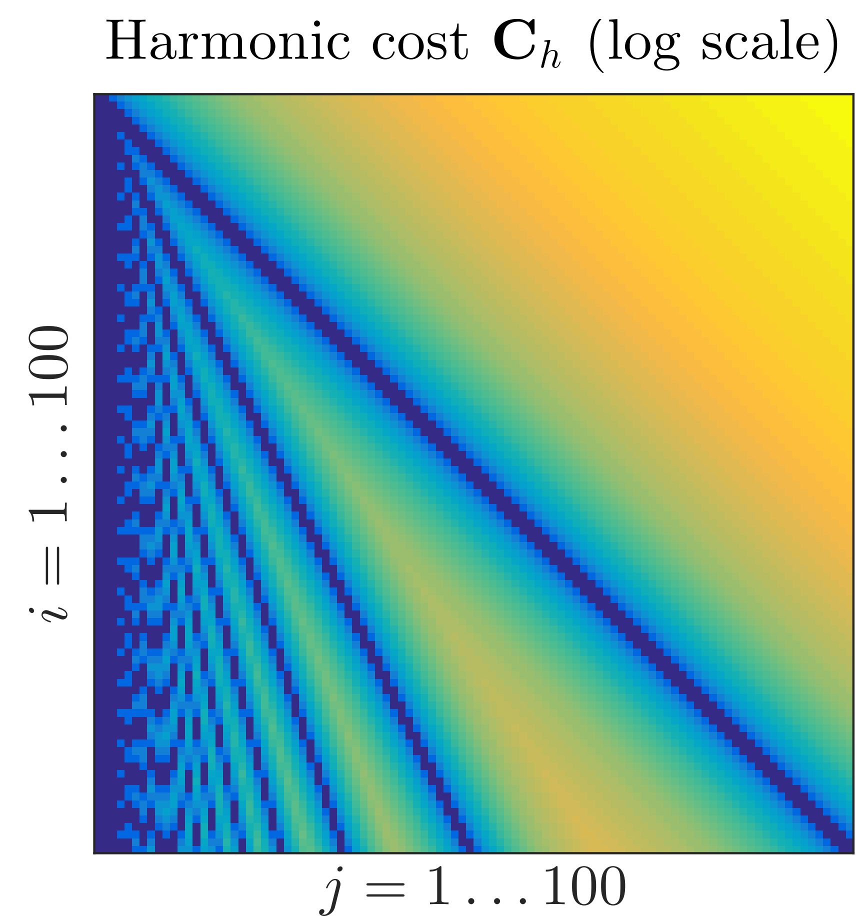
Harmonic invariant cost $\mathbf{C}_h$ defined as

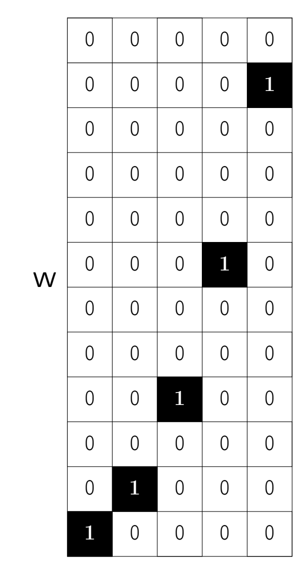
$$c_{ij} = \min_{q=1,\ldots,\left\lceil \frac{f_i}{f_j} \right\rceil} (f_i - qf_j)^2 + \epsilon\,\delta_{q\neq 1},$$

where ϵ is a small positive value.

Main features:

- term $\epsilon\,\delta_{q\neq 1}$ discriminate octaves
- dictionary $\mathbf{W}$ can be composed of diracs: $w_{ik} = \delta_{f_i=\nu_k}$, where ν_k is the fundamental frequency of the k-th note
- such a dictionary allows significant algorithmic and computational enhancements





OT unmixing with a pre-learned dictionary and quadratic cost

Original problem:

$$\min_{\mathbf{H}\geq 0} D_{\mathbf{C}}\left(\mathbf{V}\,|\mathbf{WH}\right) \;\; s.t. \;\; \forall n, \left\|\mathbf{h}_n\right\|_1 = 1$$

Using separability in time (n) and introducing the transportation matrix, it is equivalent to solve, for any n,

$$\min_{\mathbf{h}_n\geq 0, \mathbf{T}\geq 0} \langle \mathbf{T}, \mathbf{C} \rangle \;\; s.t. \;\; \begin{cases} \mathbf{T}\mathbf{1}_M & = \mathbf{v} \\ \mathbf{T}^T\mathbf{1}_M & = \mathbf{W}\mathbf{h}_n \end{cases}$$

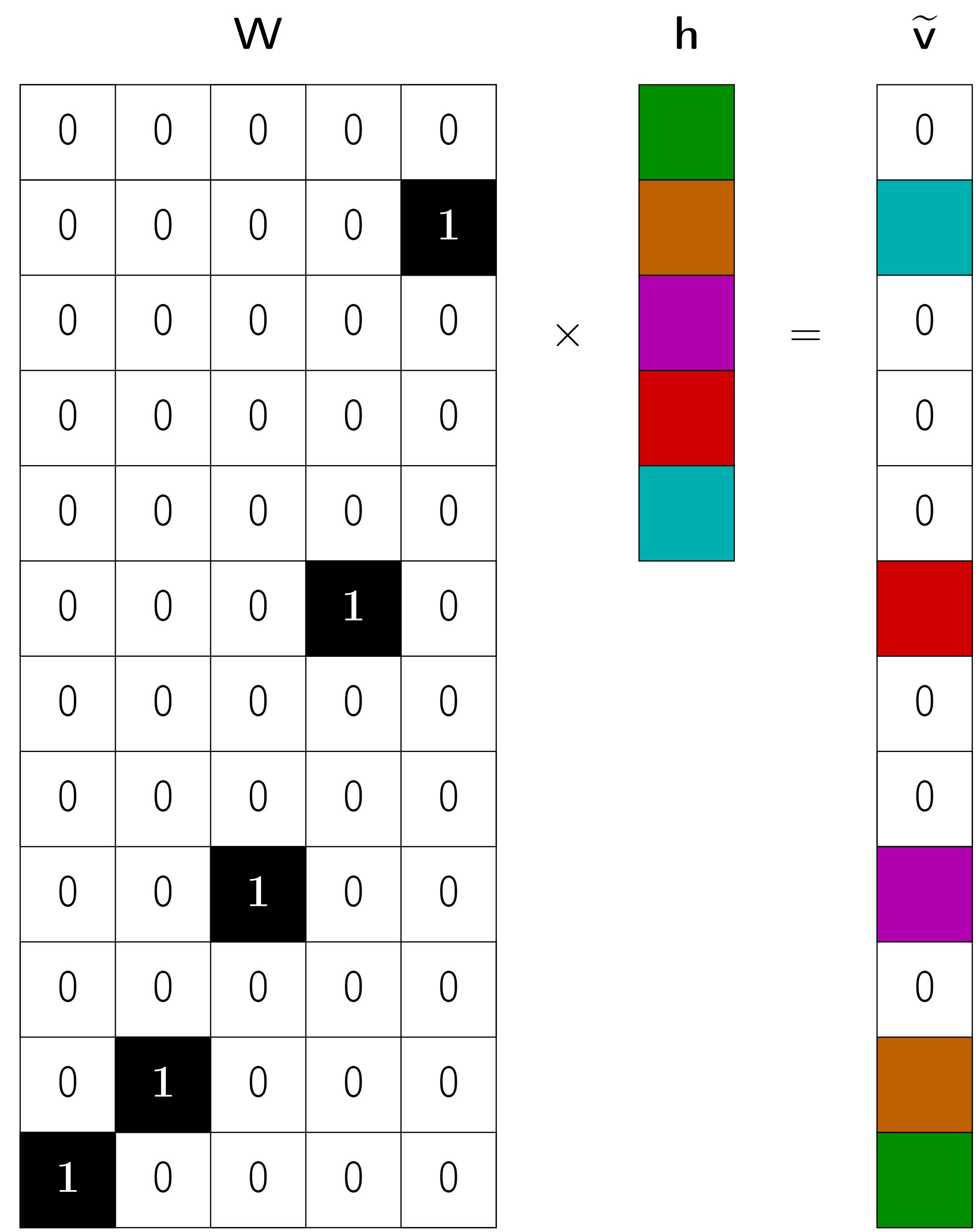
- this is a linear program
- with a large number of variables ($M^2 + K \approx 10^5$)

OT unmixing with a dirac dictionary and harmonic cost

Dimension reduction of **T** and **C**:

- $K < M$ notes in the dirac dictionary **W**
- one non-zero coefficient per column

$\Rightarrow$ $M - K$ zeros in $\widetilde{\mathbf{v}}$



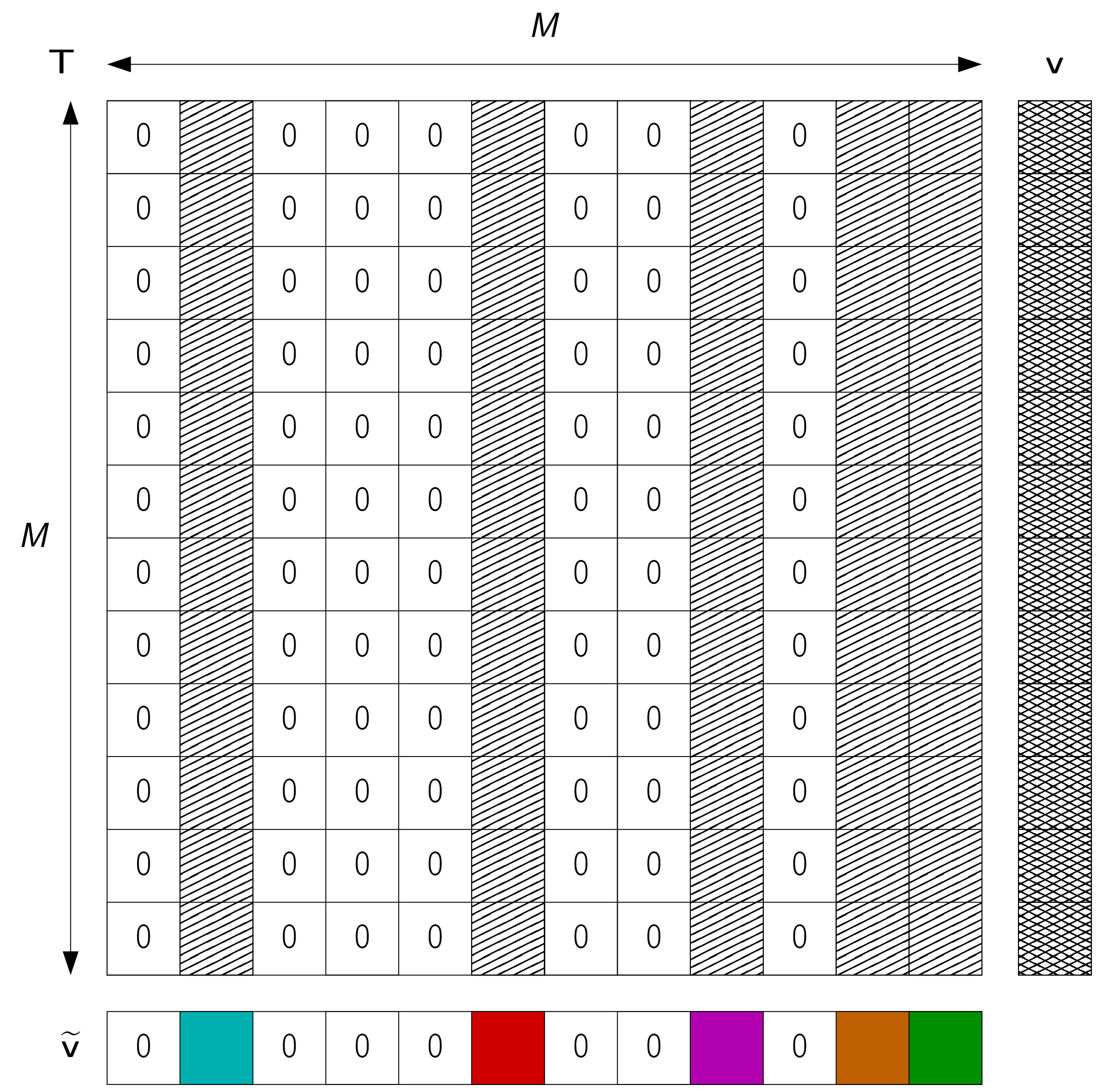
OT unmixing with a dirac dictionary and harmonic cost

Dimension reduction of **T** and **C**:

- $K < M$ notes in the dirac dictionary **W**
- one non-zero coefficient per column

$\Rightarrow$ $M - K$ zeros in $\widetilde{\mathbf{v}}$

$\Rightarrow$ zeros in related columns in **T**



OT unmixing with a dirac dictionary and harmonic cost

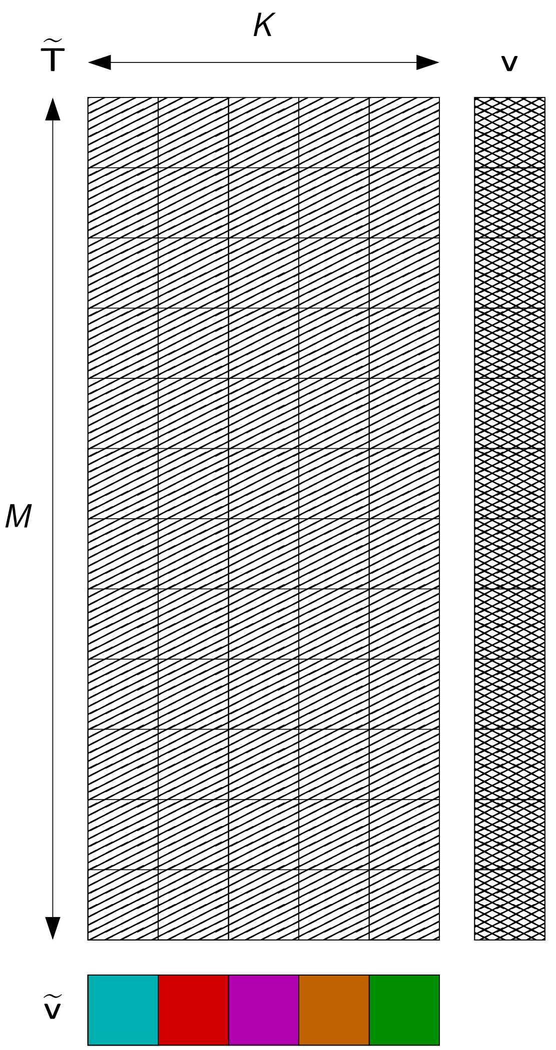
Dimension reduction of $\mathbf{T}$ and $\mathbf{C}$:

- $K < M$ notes in the dirac dictionary $\mathbf{W}$
- one non-zero coefficient per column

$\Rightarrow$ $M - K$ zeros in $\widetilde{\mathbf{v}}$

$\Rightarrow$ zeros in related columns in $\mathbf{T}$

$\Rightarrow$ $\mathbf{T}$ and $\mathbf{C}$ can be reduced to their useful columns $\widetilde{\mathbf{T}}$ and $\widetilde{\mathbf{C}}$



OT unmixing with a dirac dictionary and harmonic cost

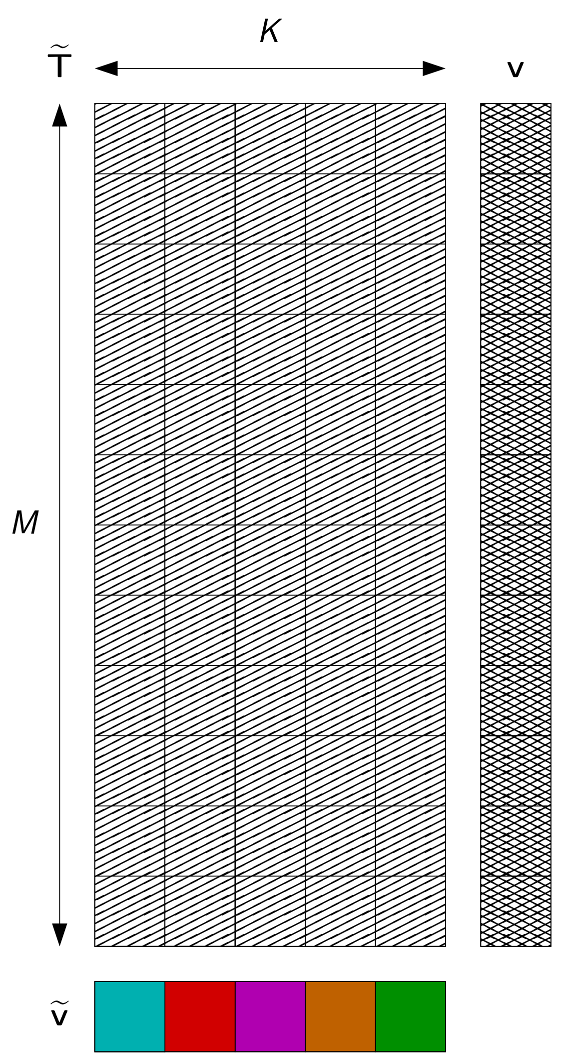
Dimension reduction of **T** and **C**:

- $K < M$ notes in the dirac dictionary $\mathbf{W}$
- one non-zero coefficient per column

$\Rightarrow$ $M - K$ zeros in $\widetilde{\mathbf{v}}$

$\Rightarrow$ zeros in related columns in $\mathbf{T}$

$\Rightarrow$ $\mathbf{T}$ and $\mathbf{C}$ can be reduced to their useful columns $\widetilde{\mathbf{T}}$ and $\widetilde{\mathbf{C}}$



Resulting problem: for any n,

$$\min_{\mathbf{h}_n \geq 0, \widetilde{\mathbf{T}} \geq 0} \left\langle \widetilde{\mathbf{T}}, \widetilde{\mathbf{C}} \right\rangle \ s.t. \ \begin{cases} \widetilde{\mathbf{T}}\mathbf{1}_K & = \mathbf{v} \\ \widetilde{\mathbf{T}}^T \mathbf{1}_M & = \mathbf{W}\mathbf{h}_n \end{cases}$$

OT unmixing with a dirac dictionary and harmonic cost

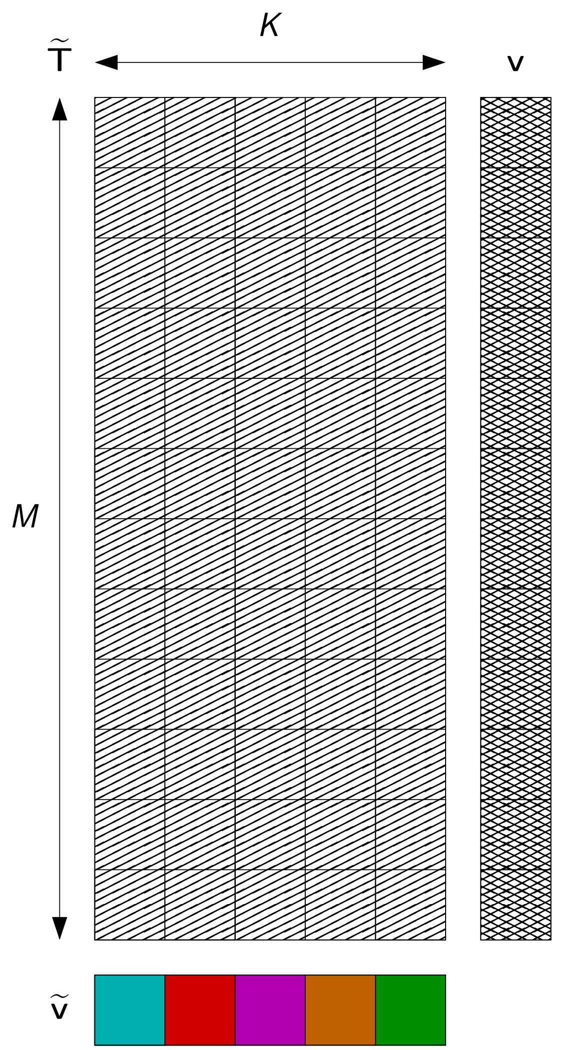
Dimension reduction of $\mathbf{T}$ and $\mathbf{C}$:

- $K < M$ notes in the dirac dictionary $\mathbf{W}$
- one non-zero coefficient per column

$\Rightarrow$ $M - K$ zeros in $\widetilde{\mathbf{v}}$

$\Rightarrow$ zeros in related columns in $\mathbf{T}$

$\Rightarrow$ $\mathbf{T}$ and $\mathbf{C}$ can be reduced to their useful columns $\widetilde{\mathbf{T}}$ and $\widetilde{\mathbf{C}}$



Resulting problem: for any n,

$$\min_{\mathbf{h}_n \geq 0, \widetilde{\mathbf{T}} \geq 0} \left\langle \widetilde{\mathbf{T}}, \widetilde{\mathbf{C}} \right\rangle \ s.t. \begin{cases} \widetilde{\mathbf{T}}\mathbf{1}_K & = \mathbf{v} \\ \widetilde{\mathbf{T}}^T \mathbf{1}_M & = \mathbf{W}\mathbf{h}_n \end{cases}$$

\+ subsequent decoupling w.r.t. the rows of $\widetilde{\mathbf{T}}$.
$\Rightarrow \mathcal{O}(M)$ (PLCA: $\mathcal{O}(KM)$ per iteration).

Adding regularisation

Entropic regularisation (OST_e):

- add penalty $\lambda \sum_{ik} \tilde{t}_{ik} \log(\tilde{t}_{ik})$
- computational complexity per frame in $\mathcal{O}(KM)$

Group regularisation (OST_g):

- add penalty $\lambda \sum_k \sqrt{\|\widetilde{\mathbf{t}}_k\|_1}$
- majoration-minimization algorithm (since no close-form solution)

Using both regularisation simultaneously is also possible.

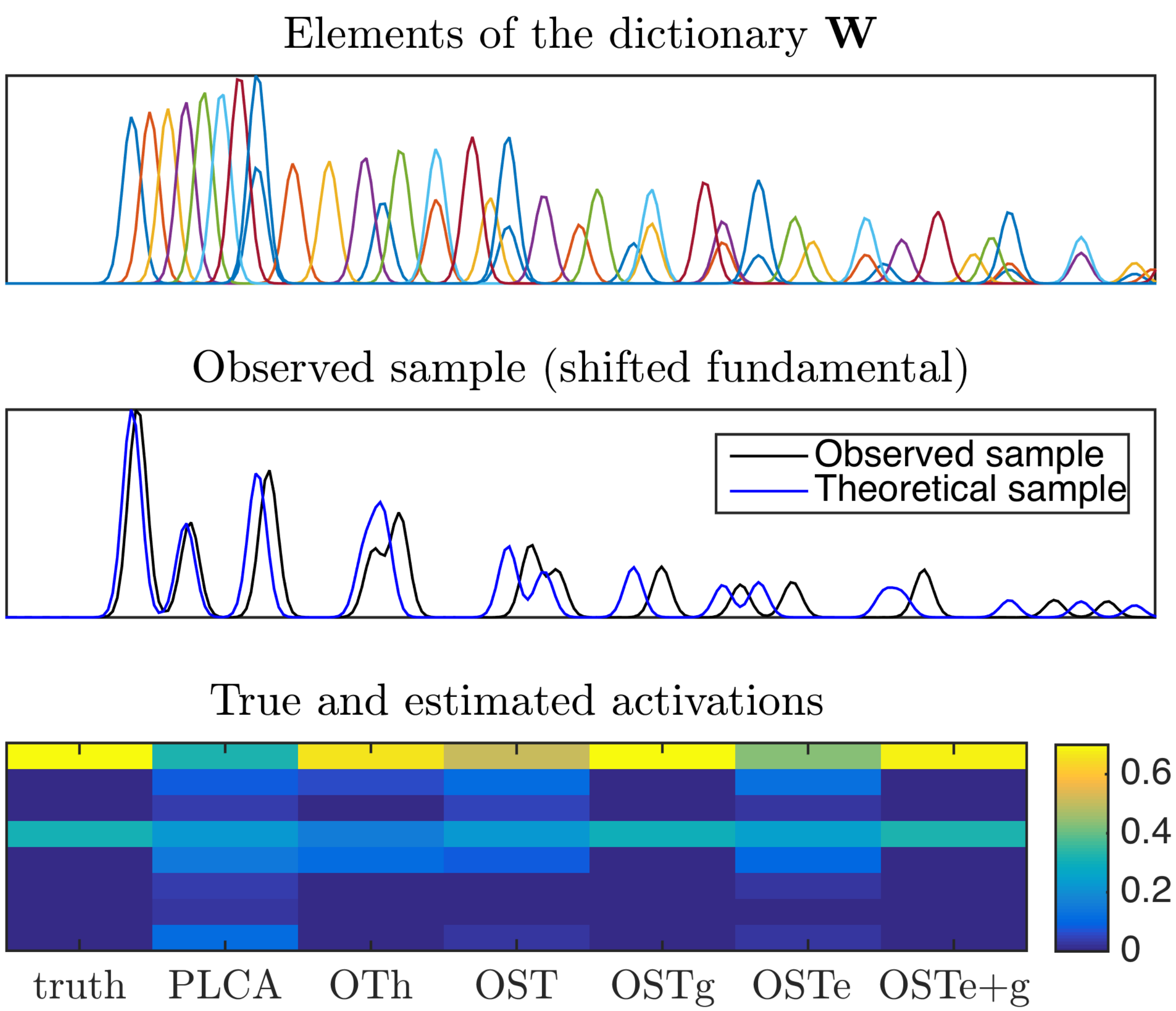
Optimal Transport for music transcription

introduction to problem
a solution with OT
some results

Toy experiments: settings

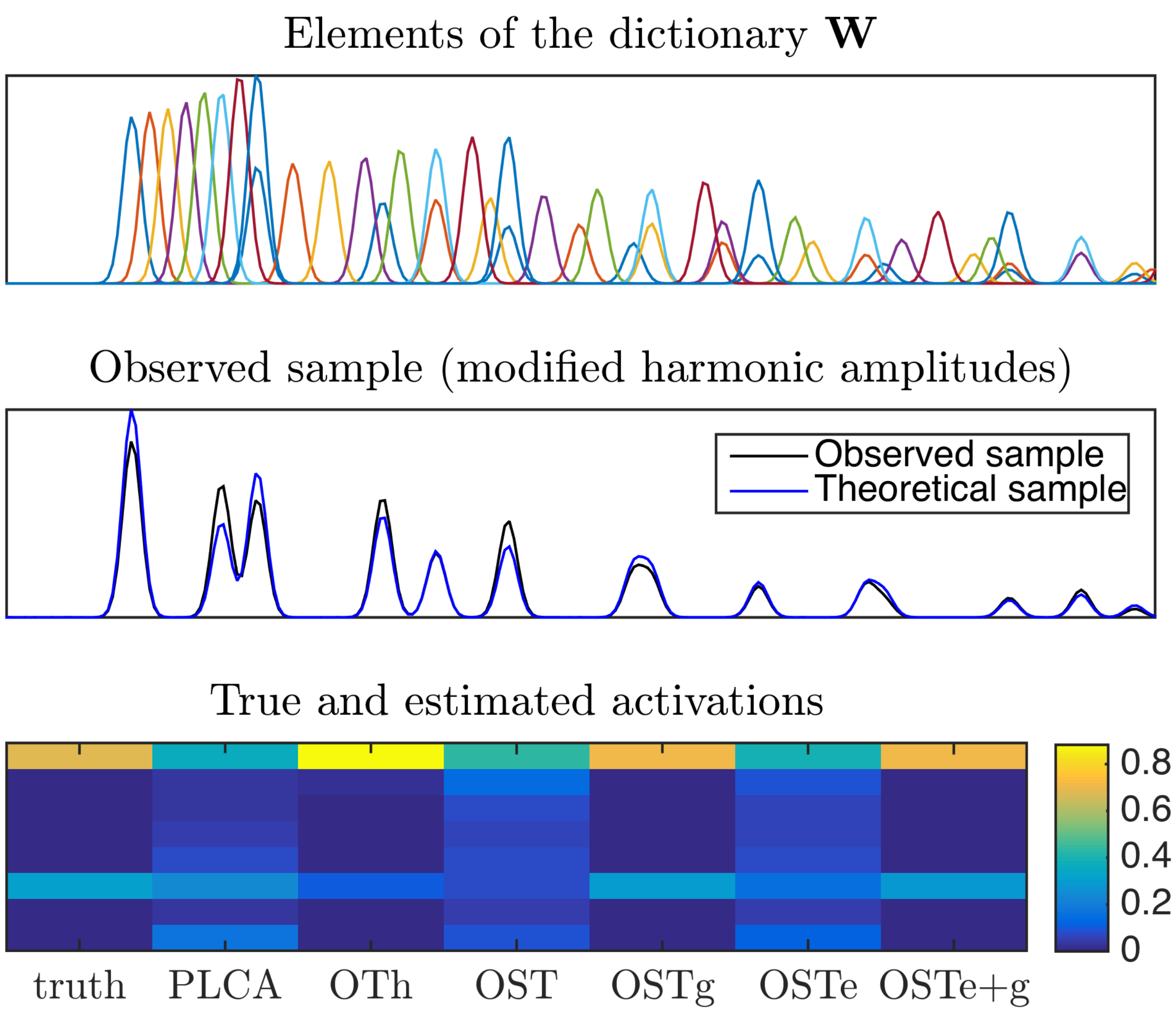
- Synthetic dictionary: 8 harmonic spectral templates with Gaussian-shape window and exponential decay in spectral envelope
- Observation 1 generated by mixing 1st and 4th components with perturbation in frequency
- Observation 2 generated by mixing 1st and 6th components with perturbation in spectral envelope
- l_1-error performance: $\left\| \widetilde{\mathbf{h}} - \mathbf{h}_{\mathbf{true}} \right\|_1$

Toy experiments: unmixing with shifted fundamental frequencies



Method	PLCA	OT_h	OST	OST_g	OST_e	OST_{e+g}
ℓ_1 error	0.900	0.340	0.534	0.021	0.660	0.015
Time (s)	0.057	6.541	0.006	0.007	0.007	0.013

Toy experiments: unmixing with wrong harmonic amplitudes



Method	PLCA	OT_h	OST	OST_g	OST_e	OST_{e+g}
ℓ_1 error	0.791	0.430	0.971	0.045	0.911	0.048
Time (s)	0.019	6.529	0.006	0.006	0.005	0.010

Transcription of real musical data: results

Recognition performance (F-measure values) and average computational unmixing times

MAPS dataset file IDs	PLCA	PLCA+noise	OST	OST+noise	OST$_e$	OST$_e$+noise
chpn_op25_e4_ENSTDkAm	0.679	0.671	0.566	0.564	**0.695**	**0.695**
mond_2_SptkBGAm	0.616	**0.713**	0.470	0.534	0.610	0.607
mond_2_SptkBGCl	0.645	0.687	0.583	0.676	0.695	**0.730**
muss_1_ENSTDkAm 4	0.613	0.478	0.513	0.550	**0.671**	0.667
muss_2_AkPnCGdD	0.587	0.574	0.531	0.611	0.667	**0.675**
mz_311_1_ENSTDkCl	0.561	0.593	0.580	0.628	0.625	**0.665**
mz_311_1_StbgTGd2	0.663	0.617	0.701	0.718	**0.747**	**0.747**
Average	0.624	0.619	0.563	0.612	0.673	**0.684**
Time (s)	14.861	15.420	**0.004**	**0.005**	0.210	0.202

Conclusions and future works

Conclusions

- OT models are able to model variability in amplitude and frequency
- does not require the design of a sofisticated dictionary
- computationally efficient solutions are provided

A Python implementation of OST and real-time demonstrator are available at

`https://github.com/rflamary/OST`

Future works

- design new cost matrices $\mathbf{C}$
- add time structure in the model
- larger experiments needed

References I

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. (2010).
A theory of learning from different domains.
Machine Learning, 79(1-2):151–175.

Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).
Iterative bregman projections for regularized transportation problems.
SIAM Journal on Scientific Computing, 37(2):A1111–A1138.

Bredies, K., Lorenz, D. A., and Maass, P. (2009).
A generalized conditional gradient method and its connection to an iterative shrinkage method.
Computational Optimization and Applications, 42(2):173–193.

Bruzzone, L. and Marconcini, M. (2010).
Domain adaptation problems: A dasvm classification technique and a circular validation strategy.
Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(5):770–787.

Courty, N., Flamary, R., and Tuia, D. (2014).
Domain adaptation with regularized optimal transport.
In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016).
Optimal transport for domain adaptation.
Pattern Analysis and Machine Intelligence, IEEE Transactions on.

References II

Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transportation.
In *NIPS*, pages 2292–2300.

Cuturi, M. and Doucet, A. (2014).
Fast computation of wasserstein barycenters.
In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014).
DeCAF: a deep convolutional activation feature for generic visual recognition.
In *Proceedings of The 31st International Conference on Machine Learning*, pages 647–655.

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014).
Regularized discrete optimal transport.
SIAM Journal on Imaging Sciences, 7(3).

Flamary, R., Févotte, C., Courty, N., and Emyia, V. (2016).
Optimal spectral transportation with application to music transcription.
In *Neural Information Processing Systems (NIPS)*.

Frogner, C., Zhang, C., Mobahi, H., Araya, M., and Poggio, T. A. (2015).
Learning with a wasserstein loss.
In *Advances in Neural Information Processing Systems*, pages 2053–2061.

References III

Germain, P., Habrard, A., Laviolette, F., and Morvant, E. (2013).
A PAC-Bayesian Approach for Domain Adaptation with Specialization to Linear Classifiers.
In *ICML*, pages 738–746, Atlanta, USA.

Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012).
Geodesic flow kernel for unsupervised domain adaptation.
In *CVPR*, pages 2066–2073. IEEE.

Hoffman, J., Rodner, E., Donahue, J., Saenko, K., and Darrell, T. (2013).
Efficient learning of domain-invariant image representations.
In *International Conference on Learning Representations*.

Kantorovich, L. (1942).
On the translocation of masses.
C.R. (Doklady) Acad. Sci. URSS (N.S.), 37:199–201.

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. (2013).
Transfer feature learning with joint distribution adaptation.
In *ICCV*, pages 2200–2207.

Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. (2014).
Transfer joint matching for unsupervised domain adaptation.
In *CVPR*, pages 1410–1417.

References IV

Monge, G. (1781).
Mémoire sur la théorie des déblais et des remblais.
De l'Imprimerie Royale.

Nakhostin, S., Courty, N., Flamary, R., Tuia, D., and Corpetti, T. (2016).
Supervised planetary unmixing with optimal transport.
In *Whorkshop on Hyperspectral Image and Signal Processing : Evolution in Remote Sensing (WHISPERS).*

Pérez, P., Gangnet, M., and Blake, A. (2003).
Poisson image editing.
ACM Trans. on Graphics, 22(3).

Perrot, M., Courty, N., Flamary, R., and Habrard, A. (2016).
Mapping estimation for discrete optimal transport.
In *Neural Information Processing Systems (NIPS).*

Perrot, M. and Habrard, A. (2015).
Regressive virtual metric learning.
In *Advances in Neural Information Processing Systems*, pages 1810–1818.

R. Gopalan, R. L. and Chellappa, R. (2014).
Unsupervised adaptation across domain shifts by generating intermediate data representations.
IEEE Trans. Pattern Analysis and Machine Intelligence, page To be published.

References V

Redko, I., Habrard, A., and Sebban, M. (2016).
Theoretical Analysis of Domain Adaptation with Optimal Transport.
ArXiv e-prints.

Rolet, A., Cuturi, M., and Peyré, G. (2016).
Fast dictionary learning with a smoothed wasserstein loss.
In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 630–638.

Rousselle, D. and Canu, S. (2015).
Optimal transport for semi-supervised domain adaptation.
In *ESANN*.

Rubner, Y., Tomasi, C., and Guibas, L. (1998).
A metric for distributions with applications to image databases.
In *ICCV*, pages 59–66.

Si, S., Tao, D., and Geng, B. (2010).
Bregman divergence-based regularization for transfer subspace learning.
IEEE Trans. Knowledge Data Eng., 22(7):929–942.

Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. (2015).
Convolutional wasserstein distances: Efficient optimal transportation on geometric domains.
ACM Transactions on Graphics (TOG), 34(4):66.

References VI

Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P. V., and Kawanabe, M. (2008).
Direct importance estimation with model selection and its application to covariate shift adaptation.
In *Advances in neural information processing systems*, pages 1433–1440.

Tuia, D., Flamary, R., Rakotomamonjy, A., and Courty, N. (2015).
Multitemporal classification without new labels: a solution with optimal transport.
In *8th International Workshop on the Analysis of Multitemporal Remote Sensing Images*.

Zen, G., Ricci, E., and Sebe, N. (2014).
Simultaneous ground metric learning and matrix factorization with earth mover's distance.
In *ICPR*, pages 3690–3695.