Optimal Transport for Machine Learning

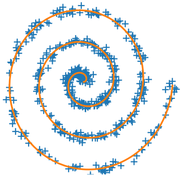
APM_52188_EP : Emerging topics in machine learning

Rémi Flamary

January 28, 2025

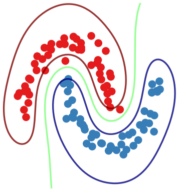
Introduction

Three aspects of Machine Learning



Unsupervised learning

- Extract information from unlabeled data
- Find labels (clustering) or subspaces/manifolds.
- Generate realistic data (GAN).

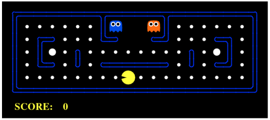


Supervised Learning

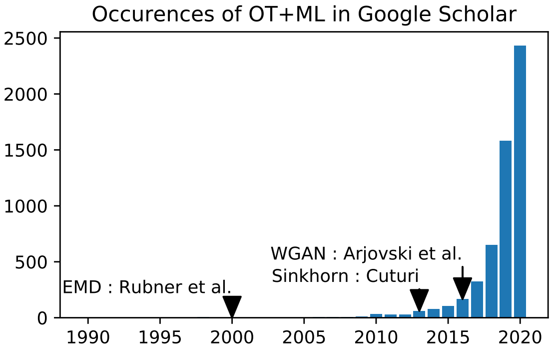
- Learning to predict from labeld dataset.
- Regression, Classification.
- Can use unsupervised information (DA, Semi-sup.)

Reinforcement Learning

- Let the machine experiment.
- Learn from its mistakes.
- Framework for learning to play games.



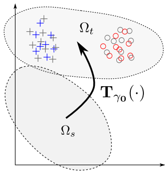
Optimal transport for machine learning

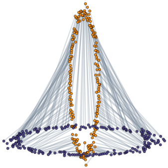


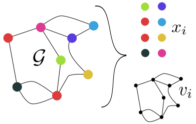
Short history of OT for ML

- Recently introduced to ML (well known in image processing since 2000s).
- Computational OT allow numerous applications (regularization).
- Deep learning boost (numerical optimization and GAN and now diffusion models).

Three aspects of optimal transport







Transporting with optimal transport

- Learn to map between distributions.
- Estimate a smooth mapping from discrete distributions.
- Applications in domain adaptation.

Divergence between histograms/empirical distributions

- Use the ground metric to encode complex relations between the bins of histograms for data fitting.
- OT losses are non-parametric divergences between non overlapping distributions.
- Used to train minimal Wasserstein estimators.

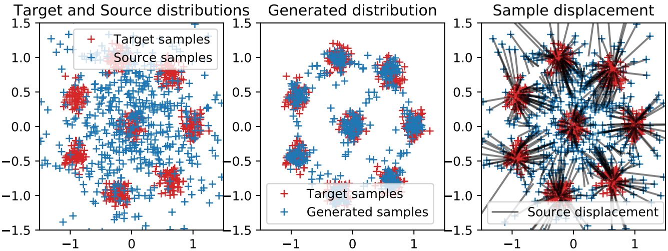
Divergence between structured objects and spaces

- Modeling of structured data and graphs as distribution.
- OT losses (Wass. or (F)GW) measure similarity between distributions/objects.
- OT find correspondance across spaces for adaptation.

Table of content

Mapping with optimal transport

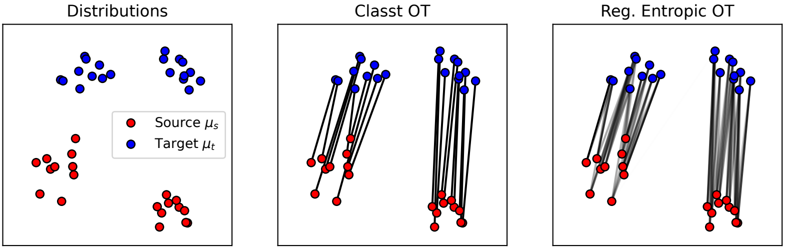
Mapping with optimal transport



Mapping estimation

- Barycentric mapping using the OT matrix [Ferradans et al., 2014].
- Linear Monge mapping when data supposed Gaussian [Flamary et al., 2019].
- Smooth mapping estimation [Perrot et al., 2016, Seguy et al., 2017, Paty et al., 2020].
- Estimation for W_2 using input convex neural networks [Makkuva et al., 2020].
- Can be used to linearize the Wasserstein space [Mérigot et al., 2020]

Transporting the discrete samples

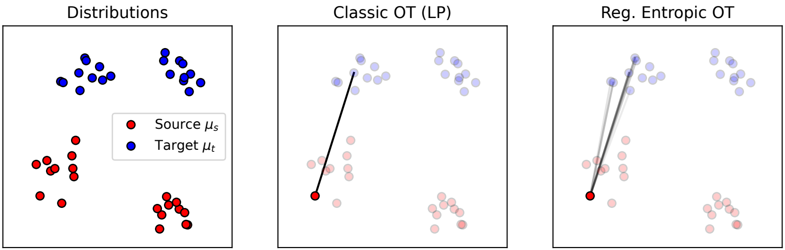


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \arg\min_{\mathbf{x}} \quad \sum_j \gamma_0(i,j)c(\mathbf{x},\mathbf{x}_j^t). \tag{1}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

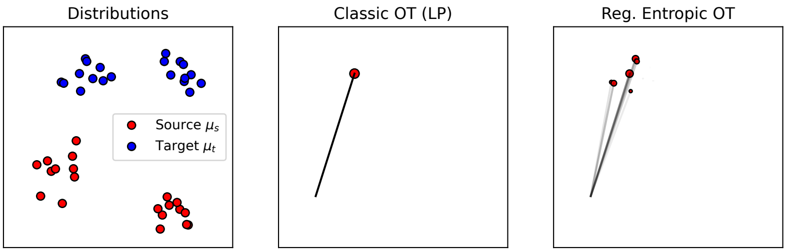


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \arg\min_{\mathbf{x}} \quad \sum_j \gamma_0(i,j)\|\mathbf{x} - \mathbf{x}_j^t\|^2. \tag{1}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

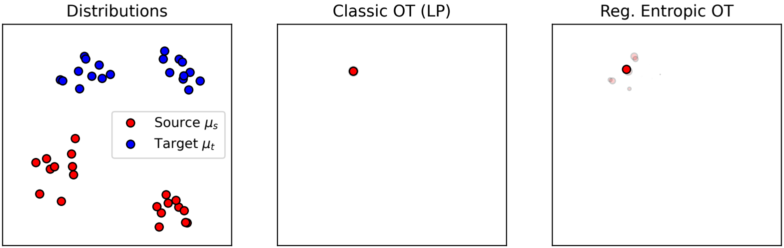


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \tag{1}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples

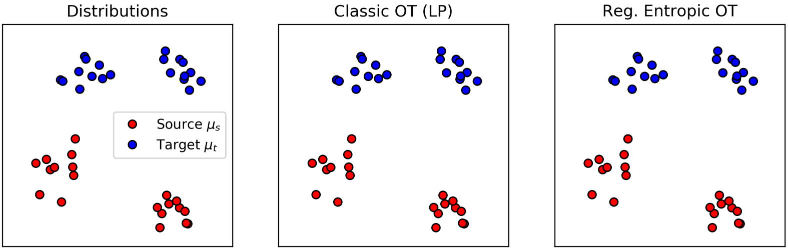


Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \tag{1}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Transporting the discrete samples



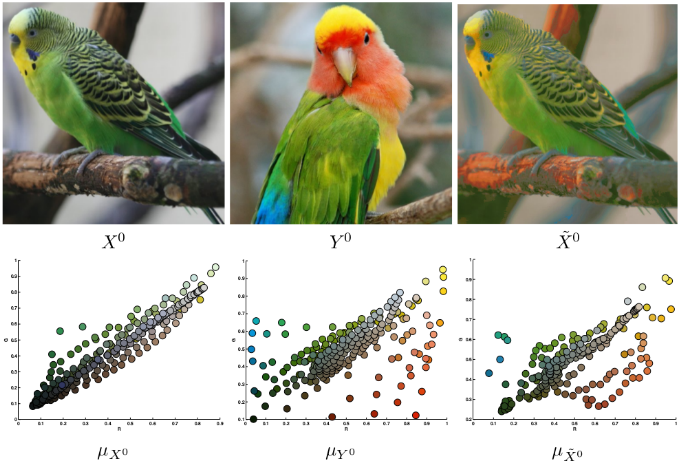
Barycentric mapping [Ferradans et al., 2014]

$$\widehat{T}_{\gamma_0}(\mathbf{x}_i^s) = \frac{1}{\sum_j \gamma_0(i,j)} \sum_j \gamma_0(i,j)\mathbf{x}_j^t. \tag{1}$$

- The mass of each source sample is spread onto the target samples (line of γ_0).
- The mapping is the barycenter of the target samples weighted by γ_0
- Closed form solution for the quadratic loss.
- Limited to the samples in the distribution (no out of sample).
- Trick: learn OT on few samples and apply displacement to the nearest point.

Histogram matching in images

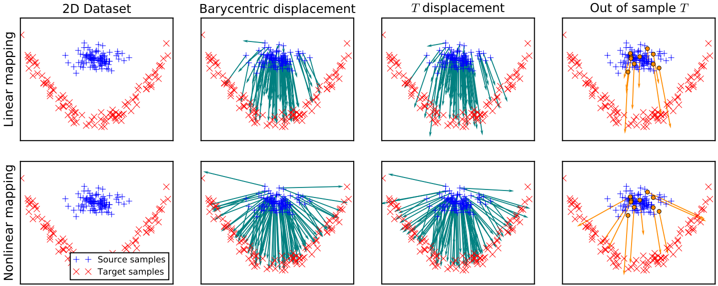
Pixels as empirical distribution [Ferradans et al., 2014]



Histogram matching in images

Image colorization [Ferradans et al., 2014]

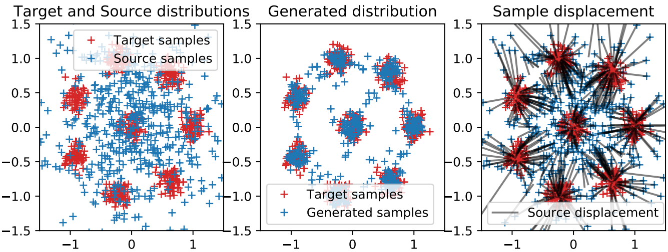
Joint OT and mapping estimation



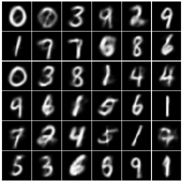
Simultaneous OT matrix and mapping [Perrot et al., 2016]

$$\min_{T, \gamma \in \mathcal{P}} \quad \langle \gamma, \mathbf{C} \rangle_F + \sum_i \|T(\mathbf{x}_i^s) - \hat{T}_\gamma(\mathbf{x}_i^s)\|^2 + \lambda \|T\|^2$$

- Estimate jointly the OT matrix and a smooth mapping approximating the barycentric mapping.
- The mapping is a regularization for OT.
- Controlled generalization error (statistical bound).
- Linear and kernel mappings T, limited to small scale datasets.

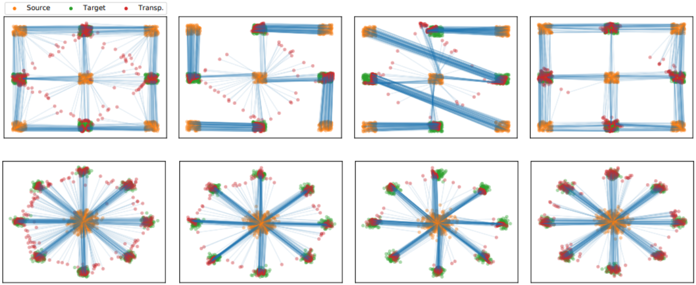


Large scale mapping estimation [Seguy et al., 2017]



- 2-step procedure:
  1. (Stochastic) estimation of regularized $\hat{\gamma}$.
  2. (Stochastic) estimation of T with a neural network.
- OT solved with Stochastic Gradient Ascent in the dual.
- Convergence to the true mapping for small regularization.
- Convergence to the smooth mapping for large n [Pooladian and Niles-Weed, 2021].

Monge Mapping with input convex neural networks

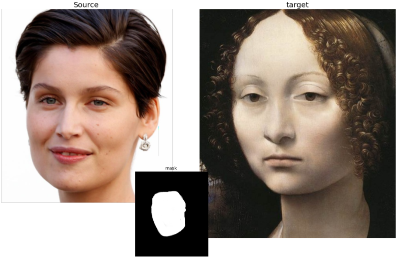


(a) Barycentric-OT (b) W1-LP (c) W2GAN (d) Our approach

Principle [Makkuva et al., 2020]

- For the quadratic cost OT between two smooth distribution Brenier theorem states that the Monge mapping is the gradient of a convex function.
- Neural network convex wrt their input (ICNN) [Amos et al., 2017].
- [Makkuva et al., 2020] proposed to estimate directly the Monge as a gradient of an ICNN from the empirical distributions.
- Conditional mappings with ICNN [Bunne et al., 2022].

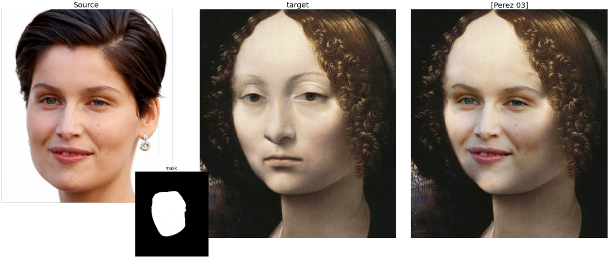
Seamless copy in images



Poisson image editing [Pérez et al., 2003]

- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy in images



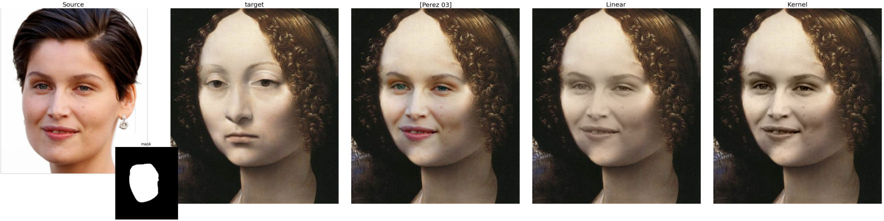
Poisson image editing [Pérez et al., 2003]

- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy with gradient adaptation [Perrot et al., 2016]

- Transport the gradient from the source to target color gradient distribution.
- Solve the Poisson equation with the mapped source gradients.
- Better respect of the color dynamic and limits false colors.

Seamless copy in images



Poisson image editing [Pérez et al., 2003]

- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy with gradient adaptation [Perrot et al., 2016]

- Transport the gradient from the source to target color gradient distribution.
- Solve the Poisson equation with the mapped source gradients.
- Better respect of the color dynamic and limits false colors.

Seamless copy in images

Poisson image editing [Pérez et al., 2003]

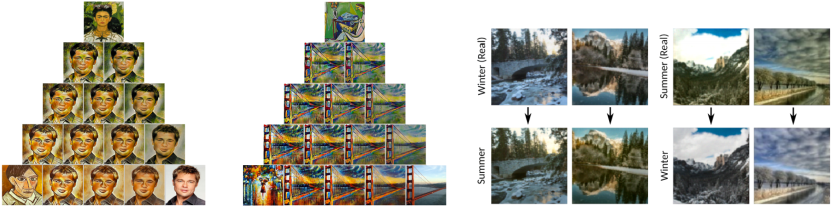
- Use the color gradient from the source image.
- Use color border conditions on the target image.
- Solve Poisson equation to reconstruct the new image.

Seamless copy with gradient adaptation [Perrot et al., 2016]

- Transport the gradient from the source to target color gradient distribution.
- Solve the Poisson equation with the mapped source gradients.
- Better respect of the color dynamic and limits false colors.

Example and webcam demo: `https://github.com/ncourty/PoissonGradient`

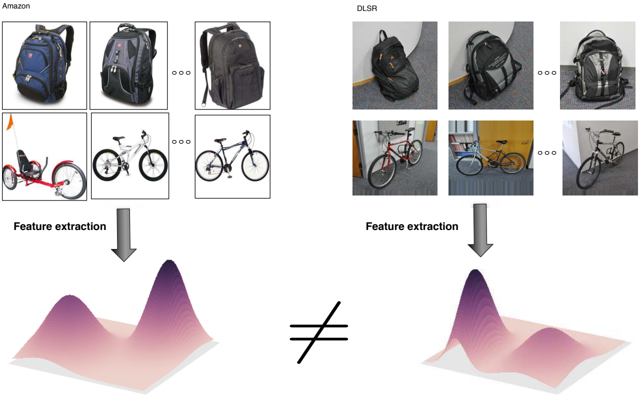
Monge mapping for Image-to-Image translation



Principle

- Encode image as a distribution in a DNN embedding.
- Transform between images using estimated Monge mapping.
- Linear Monge Mapping (Wasserstein Style Transfer [Mroueh, 2019]).
- Nonlinear Monge Mapping using input Convex Neural Networks [Korotin et al., 2019].
- Allows for transformation between two images but also style interpolation with Wasserstein barycenters.

Domain Adaptation problem

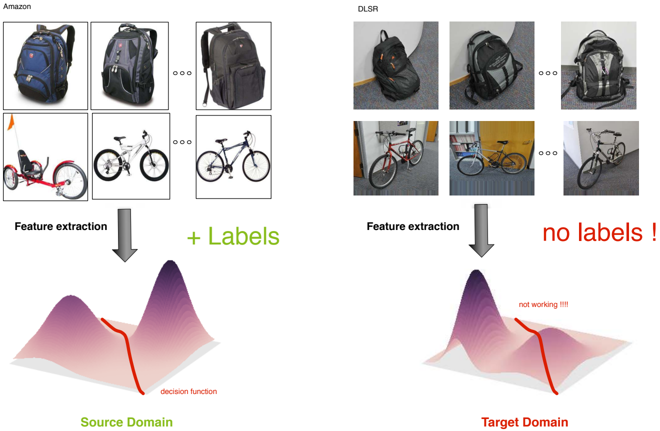


Probability Distribution Functions over the domains

Our context

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

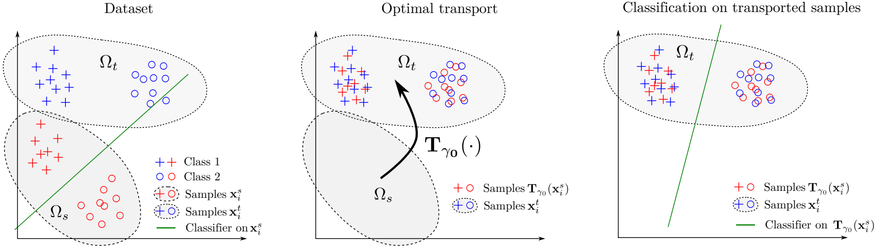
Unsupervised domain adaptation problem



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

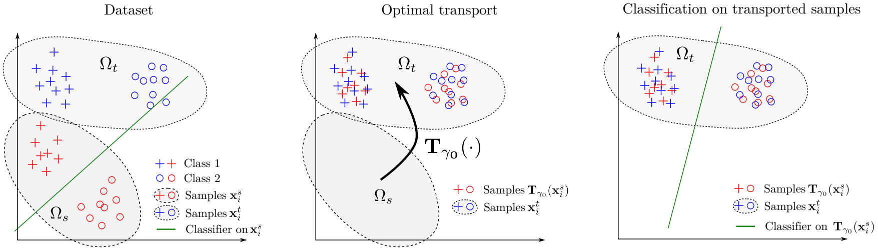
OT for domain adaptation : Step 1



Step 1 : Estimate optimal transport between distributions.

- Choose the ground metric (squared euclidean in our experiments).
- Using regularization allows
  - Large scale and regular OT with entropic regularization [Cuturi, 2013].
  - Class labels in the transport with group lasso [Courty et al., 2016].
- Efficient optimization based on Bregman projections [Benamou et al., 2015] and
  - Majoration minimization for non-convex group lasso.
  - Generalized Conditionnal gradient for general regularization (cvx. lasso, Laplacian).

OT for domain adaptation : Steps 2 & 3



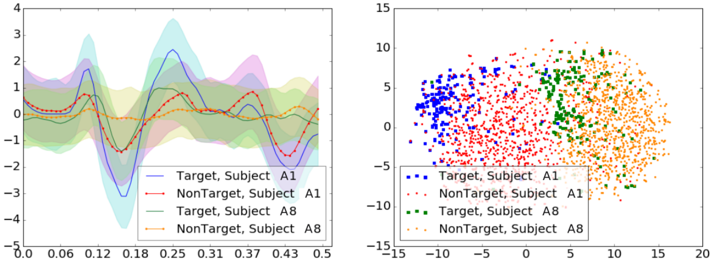
Step 2 : Transport the training samples onto the target distribution.

- The mass of each source sample is spread onto the target samples (line of γ_0).
- Transport using barycentric mapping [Ferradans et al., 2014].
- The mapping can be estimated for out of sample prediction [Perrot et al., 2016, Seguy et al., 2017].

Step 3 : Learn a classifier on the transported training samples

- Transported sample keep their labels.
- Classic ML problem when samples are well transported.

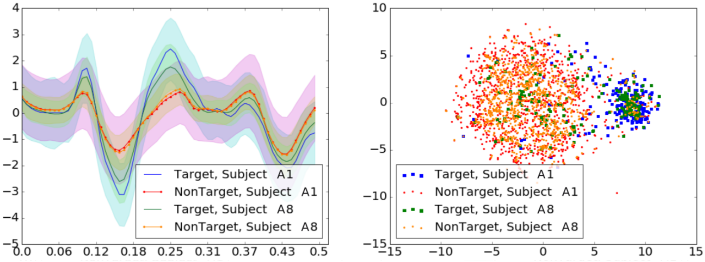
OTDA for biomedical data (1)



Multi-subject P300 classification [Gayraud et al., 2017]

- Objective : reduce calibration for BCI users.
- P300 signal is different accross subjects so adapting models is hard.
- Perform XDAWN [Rivet et al., 2009] as pre-processing.
- Use OTDA to adapt each subject in the dataset to a new subject.
- Train independent classifier on transported data and perform aggregation.

OTDA for biomedical data (1)



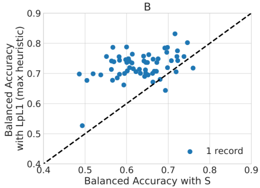
Multi-subject P300 classification [Gayraud et al., 2017]

- Objective : reduce calibration for BCI users.
- P300 signal is different accross subjects so adapting models is hard.
- Perform XDAWN [Rivet et al., 2009] as pre-processing.
- Use OTDA to adapt each subject in the dataset to a new subject.
- Train independent classifier on transported data and perform aggregation.

OTDA for biomedical data (2)

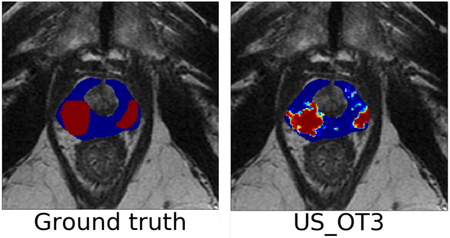
EEG sleep stage classification [Chambon et al., 2018]

- Use pre-trained neural network.
- Adapt with OTDA on the penultimate layer.
- OTDA best DA approach to adapt between EEG recordings.

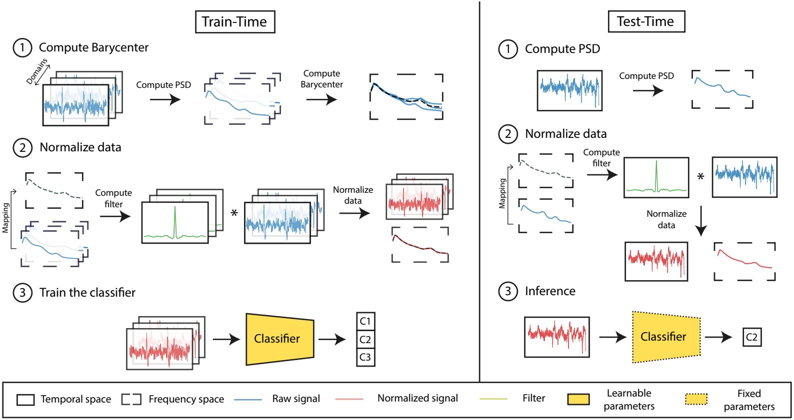


Prostace cancer classification [Gautheron et al., 2017]

- Adaptation of MRI voxel features from 1.5T to 3T.
- Achieve good performance accross subjects and modality with no target labels.

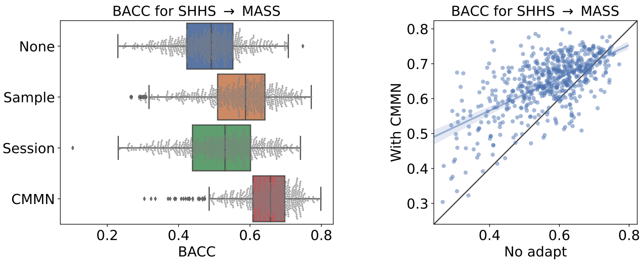


Convolutional Monge Mapping Normalization



Principle (Multi-OTDA on signal data) [Gnassounou et al., 2023]

- Multiple source datasets: compute a barycenter (Gaussian assumption).
- Map datasets to barycenter and train predictor [Montesuma and Mboula, 2021].
- At test time map test dataset to barycenter and predict.

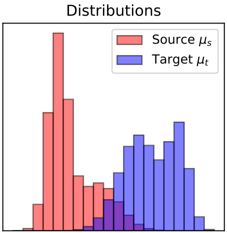


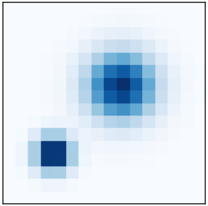
Principle (Multi-OTDA on signal data) [Gnassounou et al., 2023]

- Multiple source datasets: compute a barycenter (Gaussian assumption).
- Map datasets to barycenter and train predictor [Montesuma and Mboula, 2021].
- At test time map test dataset to barycenter and predict.
- Each domain has a specific final predictor with Mapping+Classification.
- Applied on Sleep Stage Classification problem with gain in Balanced Accuracy.
- Large gain on subjects with poor performance without adaptation.

Outline

Learning from histograms

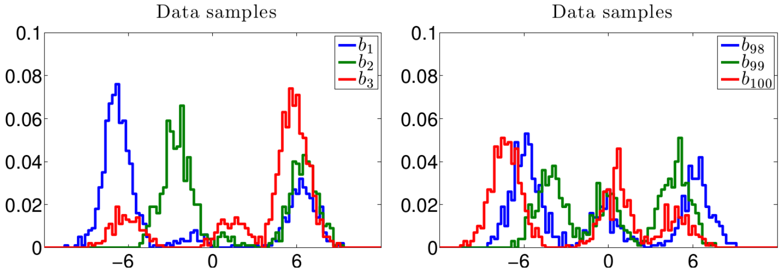




Data as histograms

- Fixed bin positions $\mathbf{x}_i$ e.g. grid, simplex $\Delta = \{(\mu_i)_i \geq 0; \sum_i \mu_i = 1\}$
- A lot of datasets comes under the form of histograms.
- Images are photo counts (black and white), text as word counts.
- Natural divergence is Kullback–Leibler.
- Not all data can be seen as histograms (positivity+constant mass)!

Dictionary learning on histograms

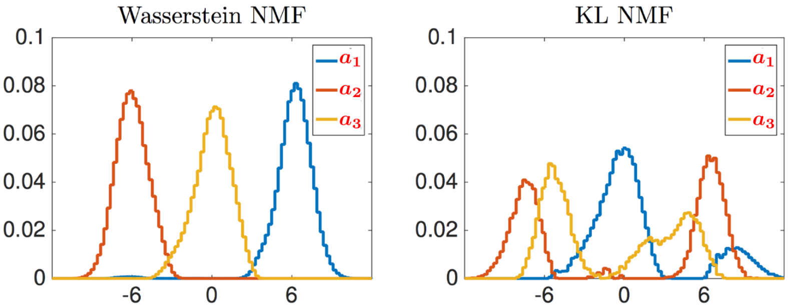


DL with Wasserstein distance [Sandler and Lindenbaum, 2011]

$$\min_{\mathbf{D},\mathbf{H}} \quad \sum_i W_{\mathbf{C}}(\mathbf{v}_i, \mathbf{D}\mathbf{h}_i)$$

- NMF: columns of $\mathbf{D}$ and $\mathbf{H}$ are on the simplex.
- Metric $\mathbf{C}$ can encode spatial relations between the bins of the histograms.
- Ground metric learning [Zen et al., 2014].
- Fast DL with regularized OT [Rolet et al., 2016].

Dictionary learning on histograms

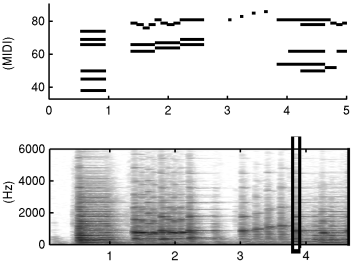


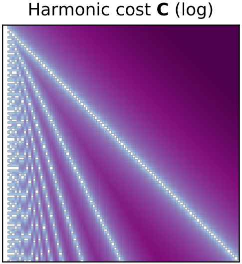
DL with Wasserstein distance [Sandler and Lindenbaum, 2011]

$$\min_{\mathbf{D},\mathbf{H}} \quad \sum_i W_{\mathbf{C}}(\mathbf{v}_i, \mathbf{D}\mathbf{h}_i)$$

- NMF: columns of $\mathbf{D}$ and $\mathbf{H}$ are on the simplex.
- Metric $\mathbf{C}$ can encode spatial relations between the bins of the histograms.
- Ground metric learning [Zen et al., 2014].
- Fast DL with regularized OT [Rolet et al., 2016].

Optimal Spectral Transportation (OST)





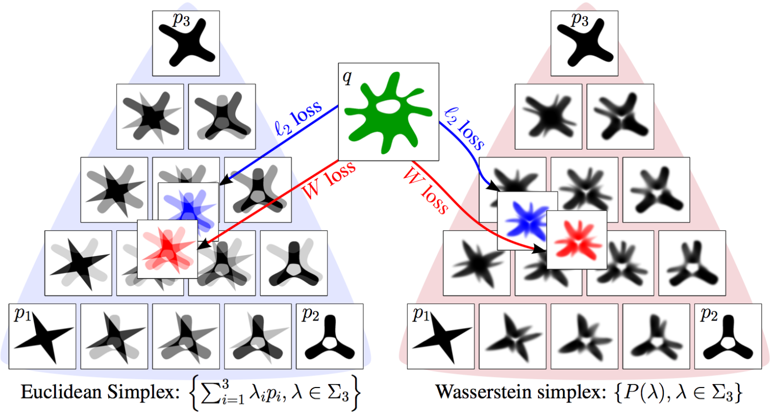
OT linear spectral unmixing of musical data [Flamary et al., 2016]

$$\min_{\mathbf{h}\in\Delta} \quad W_{\mathbf{C}}(\mathbf{v}, \mathbf{D}\mathbf{h}) \tag{2}$$

- Objective : robustness to harmonic magnitude and small frequency shift
- Encode harmonic structure in the cost matrix (harmonic robustness).
- Can use simple dictionary (diracs on fundamental frequency).
- Very fast solver for sparse and entropic regularization.

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

Wasserstein dictionary learning

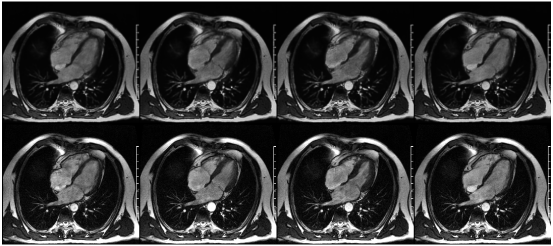


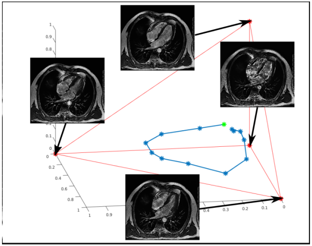
Nonlinear unmixing with Wasserstein simplex [Schmitz et al., 2017]

$$\min_{\mathbf{D},\mathbf{H}} \quad \sum_i L(\mathbf{v}_i, WB(\mathbf{D}, \mathbf{h}_i))$$

with $WB(\mathbf{D}, \mathbf{h}) = \arg\min_{\mathbf{a}} \sum_i h_i W_{\mathbf{C}}(\mathbf{d_i}, \mathbf{a})$

Wasserstein dictionary learning (2)



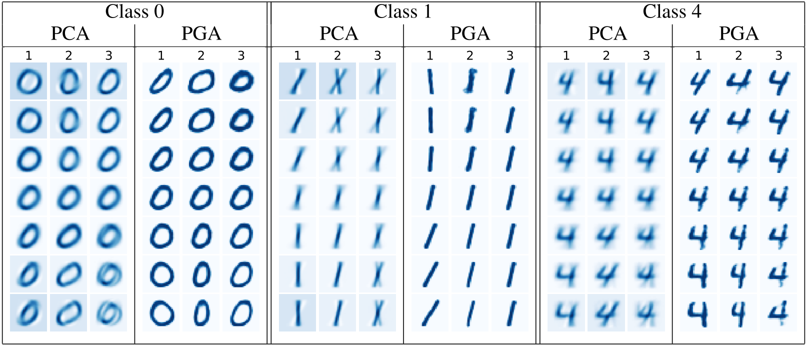


Nonlinear unmixing with Wasserstein simplex [Schmitz et al., 2017]

$$\min_{\mathbf{D},\mathbf{H}} \quad \sum_i L(\mathbf{v}_i, WB(\mathbf{D}, \mathbf{h}_i))$$

with $WB(\mathbf{D}, \mathbf{h}) = \arg\min_{\mathbf{a}} \sum_i h_i W_{\mathbf{C}}(\mathbf{d_i}, \mathbf{a})$

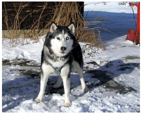
- Linear model is a barycenter for the squared ℓ_2 distance.
- Use Wasserstein barycenter for non-linear modeling.
- Application to cardiac sequence in MRI.
- One cardiac cycle is a trajectory in the simplex of the dictionary.



Geodesic PCA in the Wasserstein space [Bigot et al., 2017]

- Generalization of Principal Component Analysis to the Wassertsein manifold.
- Regularized OT [Seguy and Cuturi, 2015].
- Approximation using Wasserstein embedding [Courty et al., 2017a].

Multi-label learning with Wasserstein Loss



Siberian husky

Eskimo dog

Flickr : street, parade, dragon
Prediction : people, protest, parade

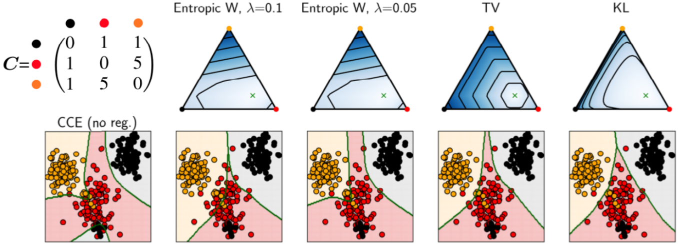
Flickr : water, boat, ref ection, sun-shine
Prediction : water, river, lake, summer;

Learning with a Wasserstein Loss [Frogner et al., 2015]

$$\min_f \quad \sum_{k=1}^{N} W_1^1(f(\mathbf{x}_i), \mathbf{l}_i)$$

- Empirical loss minimization with Wasserstein loss.
- Multi-label prediction (labels $\mathbf{l}$ seen as histograms, f output softmax).
- Cost between labels can encode semantic similarity between classes.
- Good performances in image tagging.

Wasserstein Adversarial Regularization



Principle [Fatras et al., 2021]

$$R_{\mathbf{C}}(f, \mathbf{x}) = \max_{\|\mathbf{v}\| \leq \epsilon} W_{\mathbf{C}}(f(\mathbf{x}+\mathbf{v}), f(\mathbf{x}))$$

- Use (virtual) adversarial examples to promote a better generalization of DNN (close samples should have close predictions) [Miyato et al., 2018].
- The ground metric $\mathbf{C}$ in regularization $R_{\mathbf{C}}(f, \mathbf{x})$ encodes pairwise class relations and will promote smooth/complex between them.
- State of the art performance for learning with label noise when using semantic relations between the classes for $\mathbf{C}$ (word2vec).

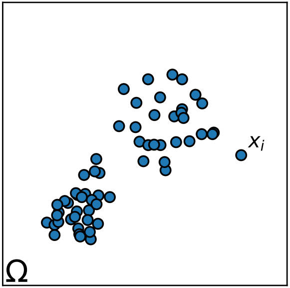
Outline

Empirical distributions A.K.A datasets

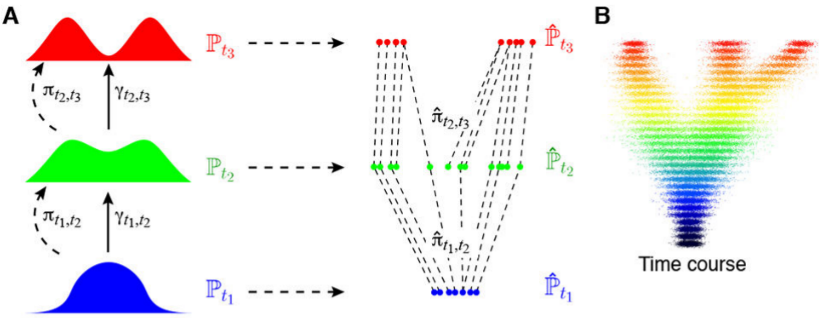
$$\mu = \sum_{i=1}^{n} a_i \delta_{\mathbf{x}_i}, \quad \mathbf{x}_i \in \Omega, \quad \sum_{i=1}^{n} a_i = 1$$

Empirical distribution

- Two realizations never overlap.
- Training base of all machine learning approaches.
- How to measure discrepancy?
- Maximum Mean Discrepancy (ℓ_2 after convolution).
- Wasserstein distance.



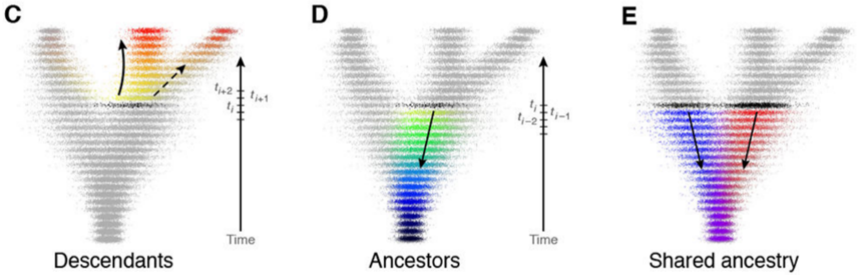
OT for modeling cell development



Principle [Schiebinger et al., 2019]

- Developmental trajectories of cells from stem cells to more specialized.
- Cell populations are samples at different times with scRNA-seq.
- Optimal transport can be used to find mapping/correspondances between across population measurements.
- Unbalanced OT is used to model cellular growth and death rates.

OT for modeling cell development



Principle [Schiebinger et al., 2019]

- Developmental trajectories of cells from stem cells to more specialized.
- Cell populations are samples at different times with scRNA-seq.
- Optimal transport can be used to find mapping/correspondances between across population measurements.
- Unbalanced OT is used to model cellular growth and death rates.
- Learning continuous version of the mapping with neural networks

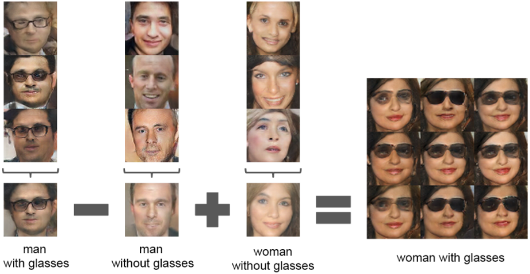
Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]

$$\min_G \max_D \quad E_{\mathbf{x}\sim\mu_d}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})}[\log(1 - D(G(\mathbf{z})))]$$

- Learn a generative model G that outputs realistic samples from data μ_d.
- Learn a classifier D to discriminate between the generated and true samples.
- Make those models compete (Nash equilibrium [Zhao et al., 2016]).

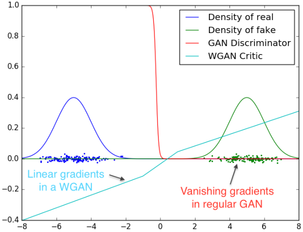
Generative Adversarial Networks (GAN)

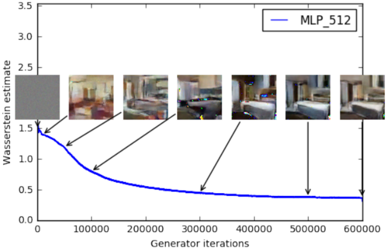


Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]

$$\min_G \max_D \quad E_{\mathbf{x}\sim\mu_d}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim\mathcal{N}(0,\mathbf{I})}[\log(1 - D(G(\mathbf{z})))]$$

- Learn a generative model G that outputs realistic samples from data μ_d.
- Learn a classifier D to discriminate between the generated and true samples.
- Make those models compete (Nash equilibrium [Zhao et al., 2016]).
- Generator space has semantic meaning [Radford et al., 2015].
- But extremely hard to train (vanishing gradients).





Wasserstein GAN [Arjovsky et al., 2017]

$$\min_{G} \quad W_1^1(G\#\mu_z, \mu_d), \tag{3}$$

- Minimizes the Wasserstein distance between the data μ_d and the generated data $G\#\mu_z$ whe $\mu_z = \mathcal{N}(0, \mathbf{I})$.
- No vanishing gradients ! Better convergence in practice.
- Wasserstein in the dual (separable w.r.t. the samples).

$$\min_{G} \sup_{\phi \in \text{Lip}^1} \quad \mathbb{E}_{\mathbf{x}\sim\mu_d}[\phi(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim\mu_z}[\phi(G(\mathbf{z}))]$$

- ϕ is a neural network that acts as an *actor critic*

WGAN: the devil in the approximation

Neural network belonging to Lip^1 ?

- Not really! [Arjovsky et al., 2017] proposes to do weight clipping that force an upper bound on the Lipschitz constant.
- It is actually the supremum over K-Lipschitz functions that is approximated by a neural network

$$\max_{f \in \text{NN class}} L_{WGAN}(f, G) \leq \sup_{\|\phi\|_L \leq K} L_{WGAN}(\phi, G) \quad = \quad K \cdot W_1^1(G(\mathbf{z}), \mu_d)$$

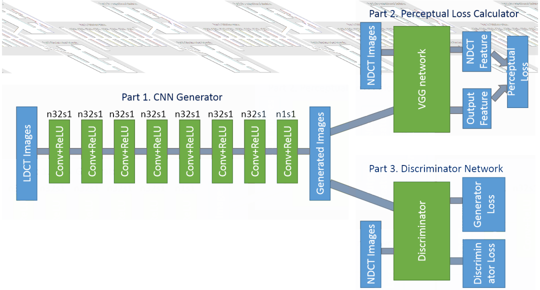
- Actually **not** equivalent to solve the optimal transport, but gradients are aligned.

Improved WGAN [Gulrajani et al., 2017]

$$\min_G \sup_{f \in \text{NN class}} \quad \mathbb{E}_{\mathbf{x}\sim\mu_d}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim\mu_z}[f(G(\mathbf{z}))] + \lambda \mathbb{E}_{\mathbf{x}\sim\mu_d}[(||\nabla f(\mathbf{x})||_2 - 1)^2]$$

Relaxation of the constraint (for W_1 the gradient of the potential is 1 almost everywhere).

Wasserstein GAN loss on Biomedical images



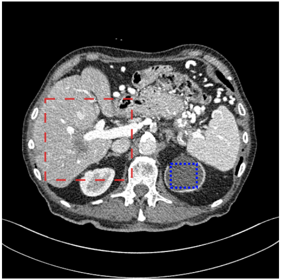
Reconstructing low dose CT images [Yang et al., 2018]

$$\min_G \quad W_1^1(G\#\mu_l, \mu_f) + \lambda_1 E_{\mathbf{x}\sim\mu_l}[\|VGG(\mathbf{x}_l) - VGG(G(\mathbf{x}_l))\|^2], \qquad (4)$$

- Use Wasserstein to make reconstruction of quarter dose CT images (μ_l) similar to high dose (resolution) CT images (μ_f).
- Perceptual loss based on VGG [Simonyan and Zisserman, 2014] embedding to keep image information.

Wasserstein GAN loss on Biomedical images

Full dose



Quarter dose

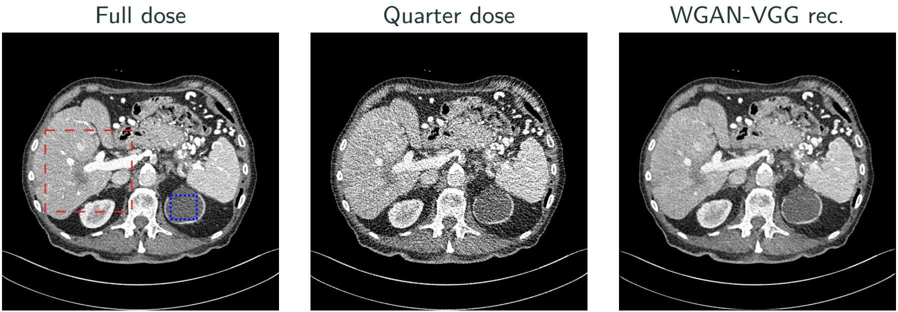
Dico rec.

Reconstructing low dose CT images [Yang et al., 2018]

$$\min_{G} \quad W_1^1(G\#\mu_l, \mu_f) + \lambda_1 E_{\mathbf{x}\sim\mu_l}[\|VGG(\mathbf{x}_l) - VGG(G(\mathbf{x}_l))\|^2], \tag{4}$$

- Use Wasserstein to make reconstruction of quarter dose CT images (μ_l) similar to high dose (resolution) CT images (μ_f).
- Perceptual loss based on VGG [Simonyan and Zisserman, 2014] embedding to keep image information.

Wasserstein GAN loss on Biomedical images

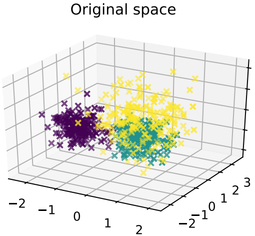


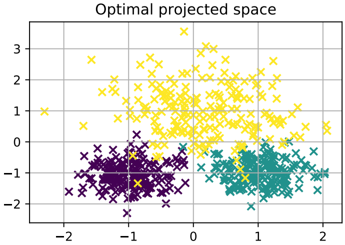
Reconstructing low dose CT images [Yang et al., 2018]

$$\min_G \quad W_1^1(G\#\mu_l, \mu_f) + \lambda_1 E_{\mathbf{x}\sim\mu_l}[\|VGG(\mathbf{x}_l) - VGG(G(\mathbf{x}_l))\|^2], \qquad (4)$$

- Use Wasserstein to make reconstruction of quarter dose CT images (μ_l) similar to high dose (resolution) CT images (μ_f).
- Perceptual loss based on VGG [Simonyan and Zisserman, 2014] embedding to keep image information.

Wasserstein Discriminant Analysis (WDA)



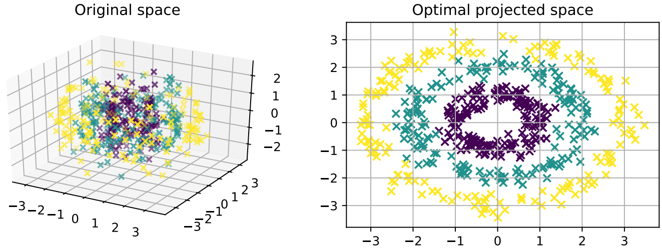


$$\max_{\mathbf{P}\in\mathcal{S}} \quad \frac{\sum_{c,c'>c} W_\lambda(\mathbf{P}\mathbf{X}^c,\mathbf{P}\mathbf{X}^{c'})}{\sum_c W_\lambda(\mathbf{P}\mathbf{X}^c,\mathbf{P}\mathbf{X}^c)} \quad (5)$$

- $\mathbf{X}^c$ are samples from class c.
- $\mathbf{P}$ is an orthogonal projection;

- Converges to Fisher Discriminant when $\lambda \to \infty$.
- Non parametric method that allows nonlinear discrimination.
- Problem solved with gradient ascent in the Stiefel manifold $\mathcal{S}$.
- Gradient computed using automatic differentiation of Sinkhorn algorithm.

Wasserstein Discriminant Analysis (WDA)

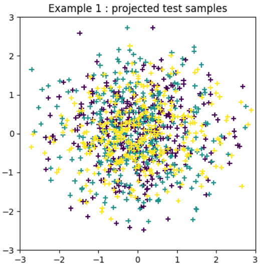


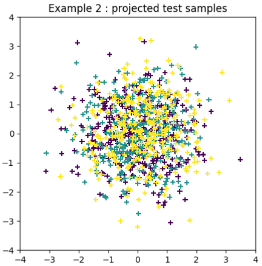
$$\max_{\mathbf{P}\in\mathcal{S}} \quad \frac{\sum_{c,c'>c} W_\lambda(\mathbf{PX}^c, \mathbf{PX}^{c'})}{\sum_c W_\lambda(\mathbf{PX}^c, \mathbf{PX}^c)} \quad (5)$$

- $\mathbf{X}^c$ are samples from class c.
- $\mathbf{P}$ is an orthogonal projection;

- Converges to Fisher Discriminant when $\lambda \to \infty$.
- Non parametric method that allows nonlinear discrimination.
- Problem solved with gradient ascent in the Stiefel manifold $\mathcal{S}$.
- Gradient computed using automatic differentiation of Sinkhorn algorithm.

Wasserstein Discriminant Analysis (WDA)



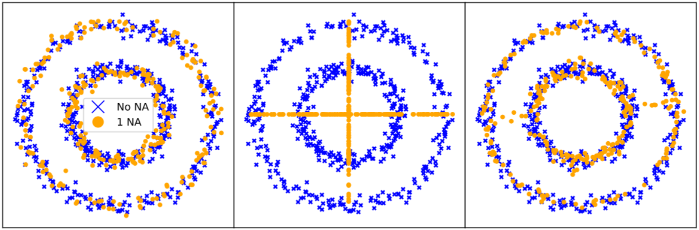


$$\max_{\mathbf{P}\in\mathcal{S}} \quad \frac{\sum_{c,c'>c} W_\lambda(\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^{c'})}{\sum_c W_\lambda(\mathbf{P}\mathbf{X}^c, \mathbf{P}\mathbf{X}^c)} \quad (5)$$

- $\mathbf{X}^c$ are samples from class c.
- $\mathbf{P}$ is an orthogonal projection;

- Converges to Fisher Discriminant when $\lambda \to \infty$.
- Non parametric method that allows nonlinear discrimination.
- Problem solved with gradient ascent in the Stiefel manifold $\mathcal{S}$.
- Gradient computed using automatic differentiation of Sinkhorn algorithm.

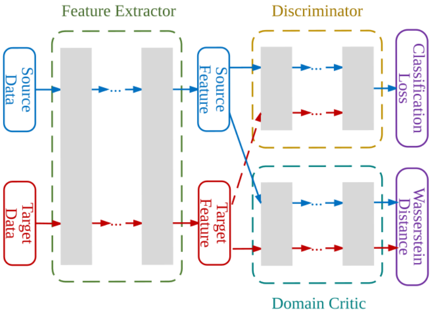
Data imputation with Optimal Transport

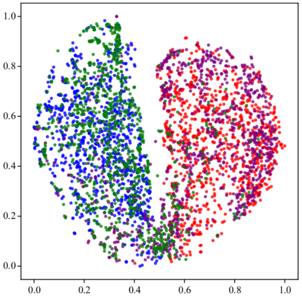


Missing Data imputation [Muzellec et al., 2020]

$$\min_{\mathbf{X}^{imp}} \quad \mathbb{E}[SD(\mu_m(\hat{\mathbf{X}}), \mu_m(\hat{\mathbf{X}}))]$$

- $\mathbf{X} \odot \mathbf{M}$ is the partially observed data with binary mask $\mathbf{M}$.
- $\hat{\mathbf{X}} = \mathbf{X} \odot \mathbf{M} + (1 - \mathbf{M}) \odot \mathbf{X}^{imp}$ is the data imputed by $\mathbf{X}^{imp}$
- $\mu_m(\mathbf{X})$ is a minibatch of $\mathbf{X}$, expectation is taken *w.r.t.* the minibatches.
- Out of sample imputation with model [Muzellec et al., 2020, Algo 2 & 3]
- Optimizing minibatch Wasserstein is a classical approach [Fatras et al., 2020].

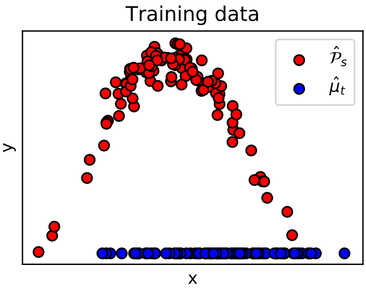


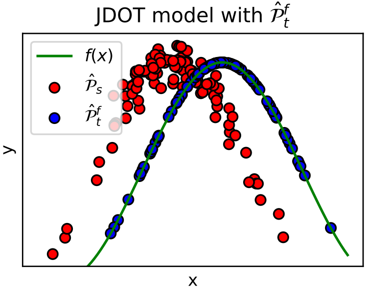


(d) t-SNE of WDGRL features

Domain adaptation for deep learning [Shen et al., 2018]

- Modern DA aim at aligning source and target in the deep representation : DANN [Ganin et al., 2016], MMD [Tzeng et al., 2014], CORAL [Sun and Saenko, 2016].
- Wasserstein distance (WGAN loss [Arjovsky et al., 2017]) used as objective for the adaptation [Shen et al., 2018].



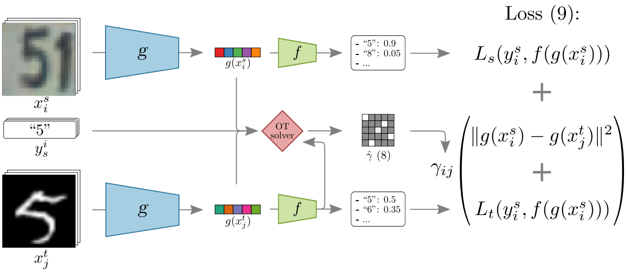


Learning with JDOT [Courty et al., 2017b]

$$\min_{f} \quad \left\{ W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t{}^f) = \inf_{\gamma \in \Pi} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, y_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) \gamma_{ij} \right\} \tag{6}$$

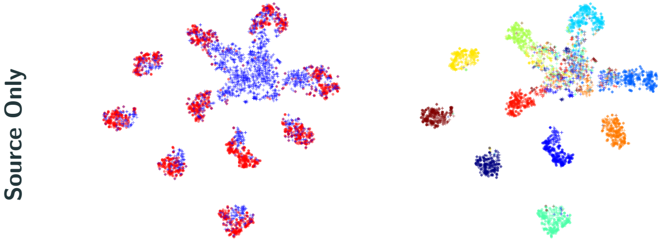
- $\hat{\mathcal{P}}_t{}^f = \frac{1}{N_t} \sum_{i=1}^{N_t} \delta_{\mathbf{x}_i^t, f\mathbf{x}_i^t}$ is the proxy joint feature/label distribution.
- $\mathcal{D}(\mathbf{x}_i^s, y_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) = \alpha \|\mathbf{x}_i^s - \mathbf{x}_j^t\|^2 + \mathcal{L}(y_i^s, f(\mathbf{x}_j^t))$ with $\alpha > 0$.
- We search for the predictor f that better align the joint distributions.
- OT matrix does the label propagation (no mapping).
- JDOT can be seen as minimizing a generalization bound.

JDOT for large scale deep learning



DeepJDOT [Damodaran et al., 2018]

- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.
- Use minibatch to estimate OT and update g, f at each iterations.
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).



DeepJDOT [Damodaran et al., 2018]

- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.
- Use minibatch to estimate OT and update g, f at each iterations.
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST→MNIST-M).

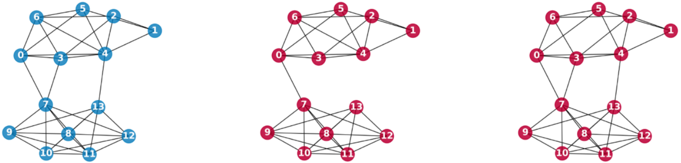
JDOT for large scale deep learning

DeepJDOT

DeepJDOT [Damodaran et al., 2018]

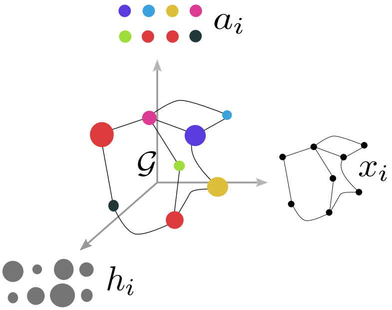
- Learn simultaneously the embedding g and the classifier f.
- JDOT performed in the joint embedding/label space.
- Use minibatch to estimate OT and update g, f at each iterations.
- Scales to large datasets and estimate a representation for both domains.
- TSNE projections of embeddings (MNIST$\rightarrow$MNIST-M).

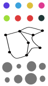
Outline



Principle [Maretic et al., 2019]

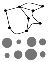
- Graph signal processing community model graph through their laplacian matrix $\mathbf{L} = \text{diag}(\mathbf{A1}) - \mathbf{A}$ where $\mathbf{A}$ is the adjacency matrix.
- The pseudo-inverse of $\mathbf{L}$ can be seen as a covariance for a Gaussian distribution for which Wasserstein has a closed form giving a similarity between graphs.
- The nodes of the two graphs are aligned by a permutation matrix that is optimized.
- Extension to graphs with different number of nodes in [Maretic et al., 2020].





$$\mu = \sum_i h_i \delta_{(x_i, a_i)}$$

$$\mu_A = \sum_i h_i \delta_{a_i}$$



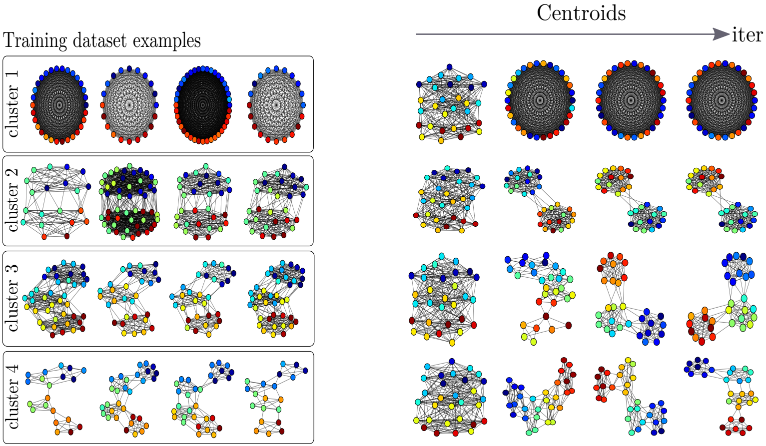
$$\mu_X = \sum_i h_i \delta_{x_i}$$

Graph data representation

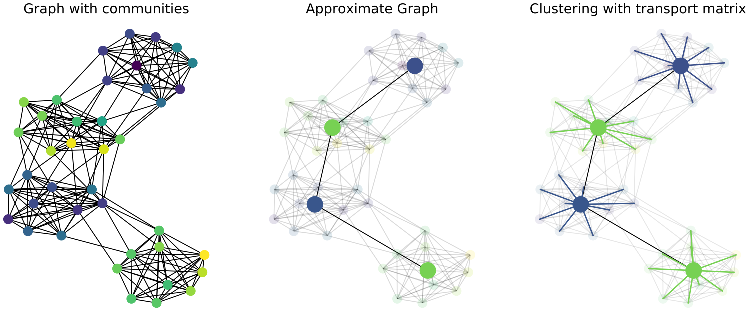
$$\mu = \sum_{i=1}^{n} h_i \delta_{(x_i a_i)}$$

- Nodes are weighted by their mass h_i.
- But no common metric between the structure points x_i of two different graphs.
- Features values a_i can be compared through the common metric
- Gromov-Wasserstein on graphs, Fused Gromov-Wasserstein on attributed graphs.

FGW for graphs based clustering



- Clustering of multiple real-valued graphs. Dataset composed of 40 graphs (10 graphs $\times$ 4 types of communities)
- k-means clustering using the FGW barycenter

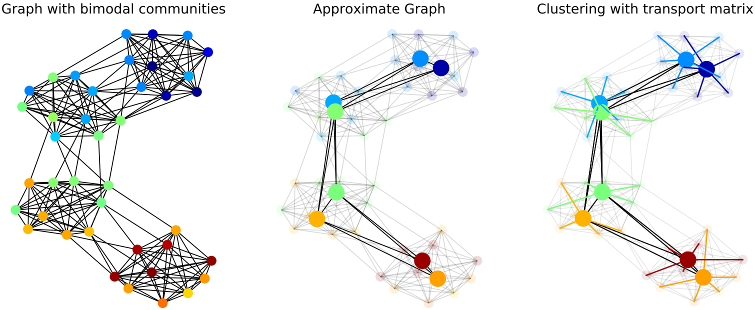


Graph approximation and comunity clustering

$$\min_{D,\mu} \quad \mathcal{FGW}(D, D_0, \mu, \mu_0)$$

- Approximate the graph (D_0, μ_0) with a small number of nodes.
- OT matrix give the clustering affectation.
- Works for signle and multiple modes in the clusters.

FGW for community clustering

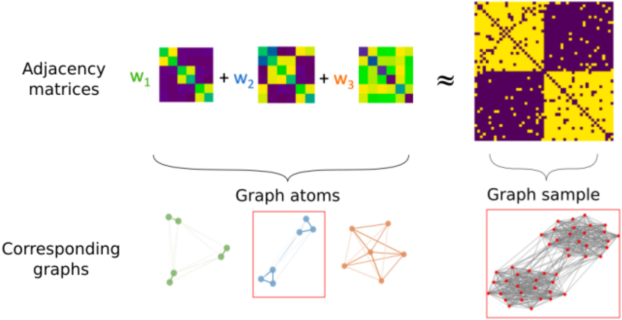


Graph approximation and comunity clustering

$$\min_{D,\mu} \quad \mathcal{FGW}(D, D_0, \mu, \mu_0)$$

- Approximate the graph (D_0, μ_0) with a small number of nodes.
- OT matrix give the clustering affectation.
- Works for signle and multiple modes in the clusters.

Linear model for graphs

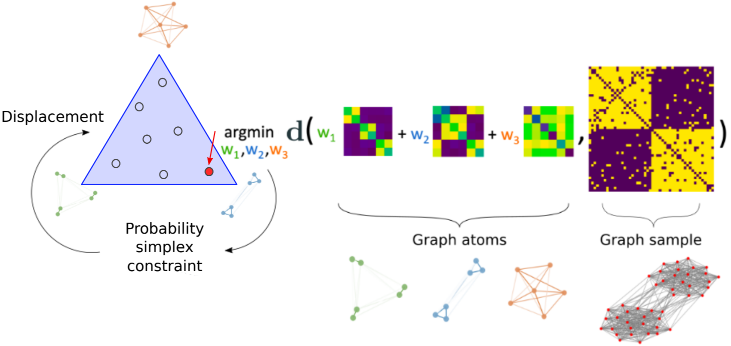


Linear modeling of graphs

$$\mathbf{C} \approx \sum_{s \in [S]} w_s \overline{\mathbf{C}_s} \tag{7}$$

- Approximate a given graph structure $\mathbf{C}$ as a non-negative weighted sum of template graphs $\overline{\mathbf{C}_s}$.
- $\{\overline{\mathbf{C}_s}\}_s$ is the dictionary of templates that all have the same order (nb. of nodes).

Gromov-Wasserstein Linear unmixing

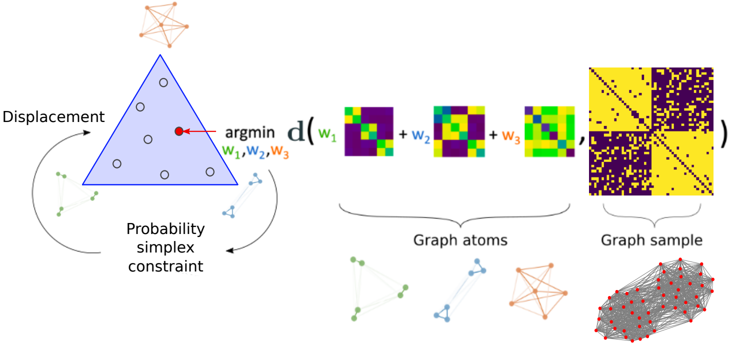


Sparse linear unmixing with Gromov-Wasserstein [Vincent-Cuaz et al., 2021]

$$\min_{\mathbf{w}\in\Sigma_S} \quad \mathcal{GW}_2^2\left(\sum_{s\in[S]} w_s \overline{\mathbf{C}_s} \ , \mathbf{C}\right) - \lambda\|\mathbf{w}\|_2^2 \tag{8}$$

- Estimate the linear representation on the simplex $\mathbf{w}$ minimizing the GW distance *w.r.t.* the target graph $\mathbf{C}$ (non-negative unmixing).
- $\lambda \in \mathbb{R}_+$, **negative quadratic regularization** promotes sparsity on the simplex [Li et al., 2016] while keeping a nonconvex QP.

Gromov-Wasserstein Linear unmixing

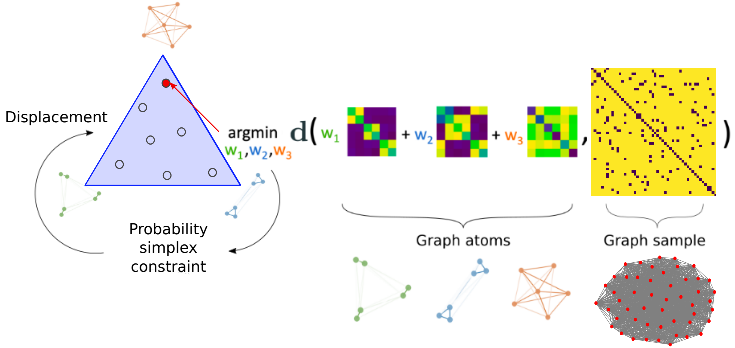


Sparse linear unmixing with Gromov-Wasserstein [Vincent-Cuaz et al., 2021]

$$\min_{\mathbf{w} \in \Sigma_S} \quad \mathcal{GW}_2^2 \left(\sum_{s \in [S]} w_s \overline{\mathbf{C}_s} \; , \; \mathbf{C} \right) - \lambda \|\mathbf{w}\|_2^2 \tag{8}$$

- Estimate the linear representation on the simplex $\mathbf{w}$ minimizing the GW distance *w.r.t.* the target graph $\mathbf{C}$ (non-negative unmixing).
- $\lambda \in \mathbb{R}_+$, **negative quadratic regularization** promotes sparsity on the simplex [Li et al., 2016] while keeping a nonconvex QP.

Gromov-Wasserstein Linear unmixing



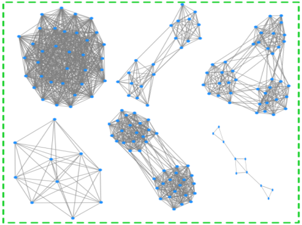
Sparse linear unmixing with Gromov-Wasserstein [Vincent-Cuaz et al., 2021]

$$\min_{\mathbf{w}\in\Sigma_S} \quad \mathcal{GW}_2^2\left(\sum_{s\in[S]} w_s \overline{\mathbf{C}_s}\ ,\ \mathbf{C}\right) - \lambda\|\mathbf{w}\|_2^2 \tag{8}$$

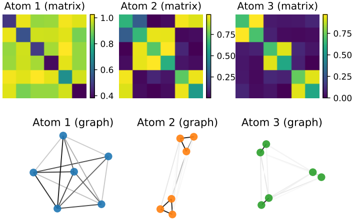
- Estimate the linear representation on the simplex $\mathbf{w}$ minimizing the GW distance *w.r.t.* the target graph $\mathbf{C}$ (non-negative unmixing).
- $\lambda \in \mathbb{R}_+$, **negative quadratic regularization** promotes sparsity on the simplex [Li et al., 2016] while keeping a nonconvex QP.

Gromov-Wassrestein dictionary learning

Dataset



Learned atoms



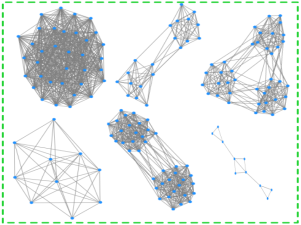
Graph Dictionary learning [Vincent-Cuaz et al., 2021]

$$\min_{\substack{\{\mathbf{w}^{(k)}\}_{k\in[K]} \\ \{\overline{\mathbf{C}}_s\}_{s\in[S]}}} \sum_{k=1}^{K} \mathcal{GW}_2^2\left(\mathbf{C}^{(k)}, \sum_{s\in[S]} w_s^{(k)}\overline{\mathbf{C}}_s\right) - \lambda\|\mathbf{w}^{(k)}\|_2^2 \quad (9)$$

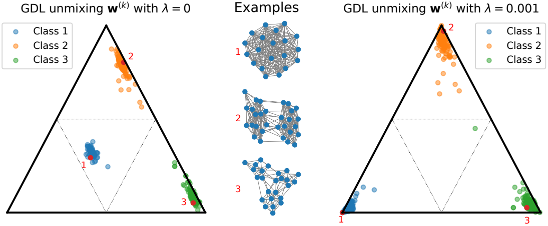
- On a dataset of K undirected graphs $\{\mathbf{C}^{(k)} \in S_{N^{(k)}}(\mathbb{R})\}_{k\in[K]}$.
- We want to estimate simultaneously the unmixing $\mathbf{w}^{(k)}$ of each graphs and the optimal dictionary $\{\overline{\mathbf{C}}_s\}_{s\in[S]}$.
- Very similar to classical DL approach but with GW as a data fitting term.

Gromov-Wassrestein dictionary learning

Dataset



Embedding space

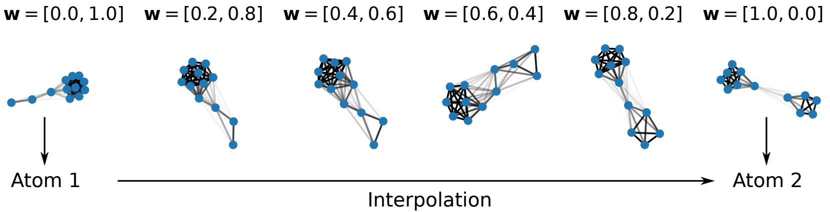


Graph Dictionary learning [Vincent-Cuaz et al., 2021]

$$\min_{\substack{\{\mathbf{w}^{(k)}\}_{k\in[K]} \\ \{\overline{\mathbf{C}}_s\}_{s\in[S]}}} \sum_{k=1}^{K} \mathcal{GW}_2^2\left(\mathbf{C}^{(k)}, \sum_{s\in[S]} w_s^{(k)} \overline{\mathbf{C}}_s\right) - \lambda \|\mathbf{w}^{(k)}\|_2^2 \qquad (9)$$

- On a dataset of K undirected graphs $\{\mathbf{C}^{(k)} \in S_{N^{(k)}}(\mathbb{R})\}_{k\in[K]}$.
- We want to estimate simultaneously the unmixing $\mathbf{w}^{(k)}$ of each graphs and the optimal dictionary $\{\overline{\mathbf{C}}_s\}_{s\in[S]}$.
- Very similar to classical DL approach but with GW as a data fitting term.

Gromov-Wassrestein dictionary learning

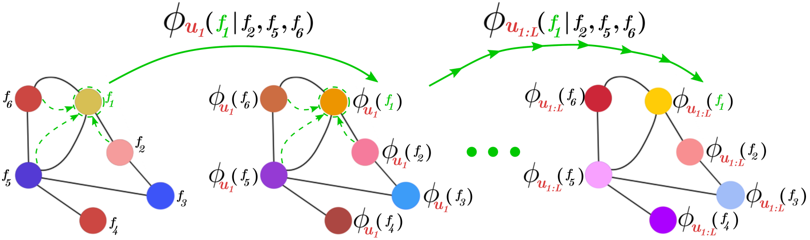


Graph Dictionary learning [Vincent-Cuaz et al., 2021]

$$\min_{\substack{\{\mathbf{w}^{(k)}\}_{k\in[K]}\\ \{\overline{\mathbf{C}}_s\}_{s\in[S]}}} \sum_{k=1}^{K} \mathcal{GW}_2^2\left(\mathbf{C}^{(k)}, \sum_{s\in[S]} w_s^{(k)}\overline{\mathbf{C}}_s\right) - \lambda\|\mathbf{w}^{(k)}\|_2^2 \tag{9}$$

- On a dataset of K undirected graphs $\{\mathbf{C}^{(k)} \in S_{N^{(k)}}(\mathbb{R})\}_{k\in[K]}$.
- We want to estimate simultaneously the unmixing $\mathbf{w}^{(k)}$ of each graphs and the optimal dictionary $\{\overline{\mathbf{C}}_s\}_{s\in[S]}$.
- Very similar to classical DL approach but with GW as a data fitting term.

Graph Neural Networks

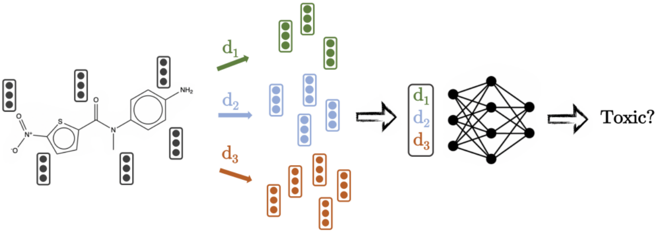


Principle [Bronstein et al., 2017]

- Each layer of the GNN compute features on graph node using the values from the connected neighbors : message passing principle.
- A step of global aggregation or pooling allows to go from a complex graph object to a vector representation.
- The pooling step must remain invariant to permutations (min, max, mean).

Can we encode graphs as disributions in GNN?

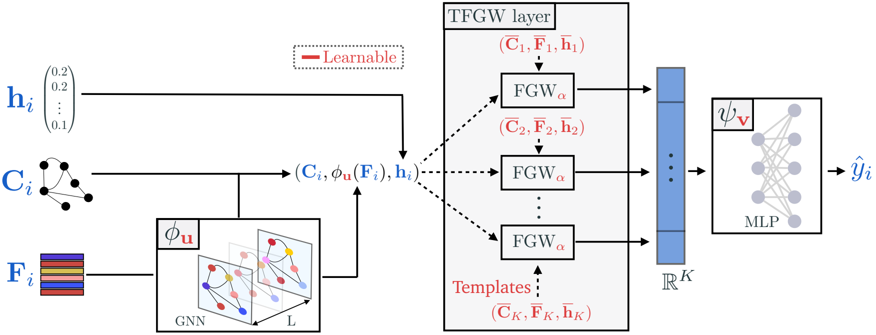
Wasserstein on Graph Convolutional Networks embeddings



Principle [Bécigneul et al., 2020]

- Extract structural features features the nodes of the graph using a Convolutional Graph neural Network.
- Models the nodes as samples of an empirical distribution (permutation invariance).
- Compute Wasserstein distance between the input graph and learned template distributions and use this as features for a final multi layer neural network.
- Diffusion Wasserstein is a linear alternative to GCN for similarity between graphs [Barbe et al., 2020].

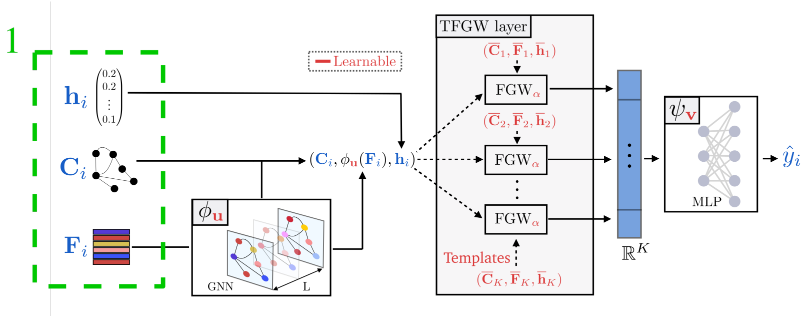
Template based Graph Neural Network with OT Distances



Template based FGW layer (TFGW) [Vincent-Cuaz et al., 2022]

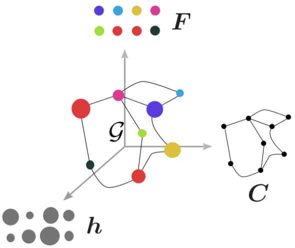
- Principle: represent a graph through its distances to learned templates.
- Novel pooling layer derived from OT distances.
- New end-to-end GNN models for graph-level tasks.
- Learnable parameters are illustrated in red above.

Template based Graph Neural Network with OT Distances

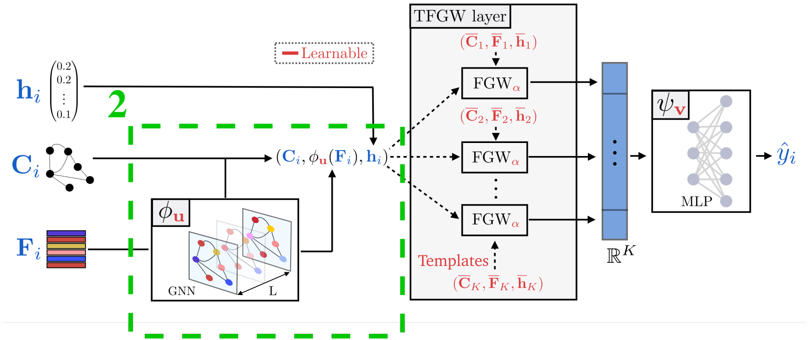


1. Modeling graphs as discrete distributions

- $\mathbf{C}_i$: node relationship matrix *e.g* adjacency, shortest-path, laplacian, etc.
- $\mathcal{F}_i$: node feature matrix.
- $\mathbf{h}_i$: nodes relative importance (probabilities).

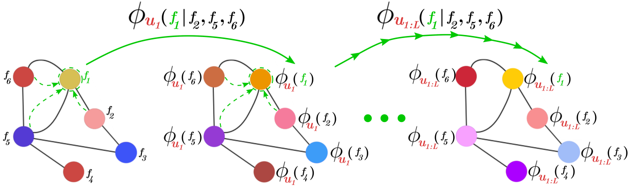


Template based Graph Neural Network with OT Distances

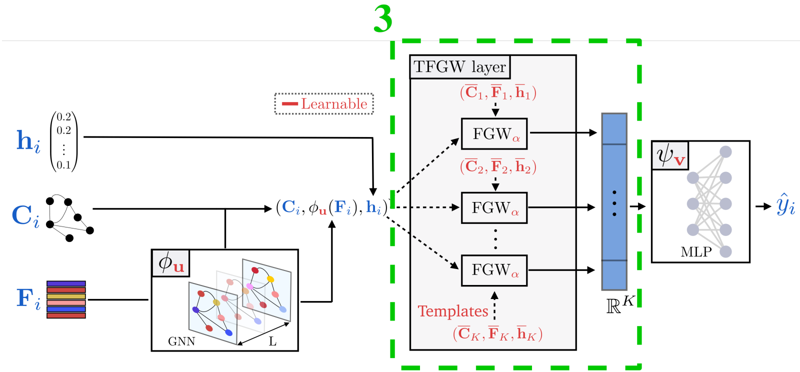


2. Node embeddings

- $\phi_{\mathbf{u}}$: GNN of L layers parameterized by $\mathbf{u}$ *e.g* GIN, GAT, etc.
- Promotes discriminant features on the nodes $\phi_{\mathbf{u}}(\mathcal{F}_i)$

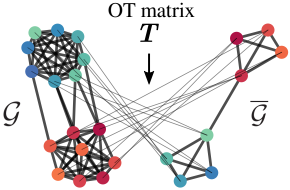


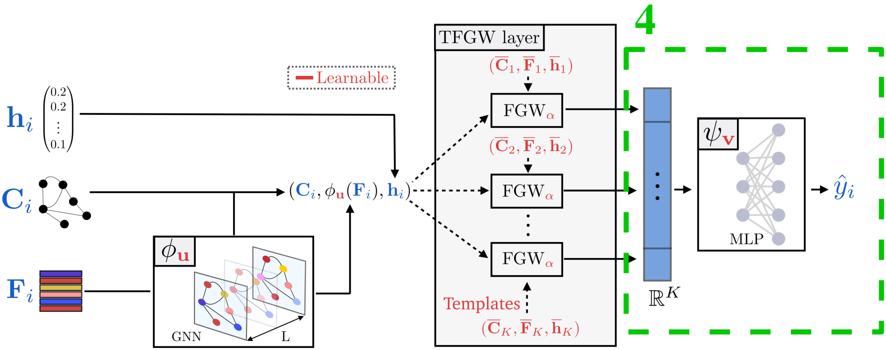
Template based Graph Neural Network with OT Distances



3. Template-based Fused Gromov-Wasserstein (TFGW) pooling

- FGW_α: OT soft graph matching distance.
- $\alpha \in [0;1]$: relative importance between structure $\mathbf{C}_i$ and node features $\phi_{\mathbf{u}}(\mathcal{F}_i)$.
- $\{\overline{\mathbf{C}}_k, \overline{\mathcal{F}}_k, \overline{\mathbf{h}}_k\}$: FGW distances to K templates used as graph representation.





4. Final MLP for predictions

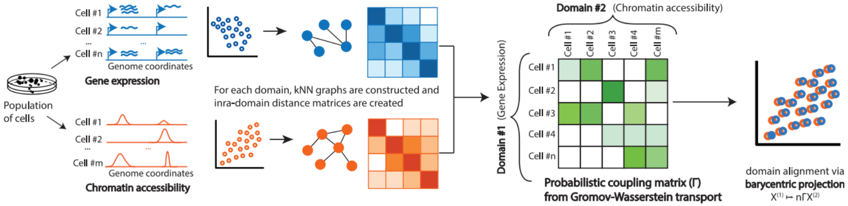
- $\psi_\mathbf{v}$: MLP with non-linearities fed with the distance embeddings.
- $\hat{y}_i$: final prediction for graph-level tasks (classification or regression).
- End-to-end optimization of all parameters:
  - $\mathbf{u}$ and $\mathbf{v}$ parameters of GNN $\phi_\mathbf{u}$ and final MLP $\psi_\mathbf{v}$.
  - $\{\overline{\mathbf{C}}_k, \overline{\mathcal{F}}_k, \overline{\mathbf{h}}_k\}$ TFGW graph templates.

TFGW benchmark

category	model	MUTAG	PTC	ENZYMES	PROTEIN	NCI1	IMDB-B	IMDB-M	COLLAB
Ours	TFGW ADJ (L=2)	**96.4(3.3)**	**72.4(5.7)**	73.8(4.6)	**82.9(2.7)**	**88.1(2.5)**	**78.3(3.7)**	**56.8(3.1)**	**84.3(2.6)**
($\phi_{\mathbf{u}}$ = GIN)	TFGW SP (L=2)	94.8(3.5)	70.8(6.3)	**75.1(5.0)**	82.0(3.0)	86.1(2.7)	74.1(5.4)	54.9(3.9)	80.9(3.1)
OT emb.	OT-GNN (L=2)	91.6(4.6)	68.0(7.5)	66.9(3.8)	76.6(4.0)	82.9(2.1)	67.5(3.5)	52.1(3.0)	80.7(2.9)
	OT-GNN (L=4)	92.1(3.7)	65.4(9.6)	67.3(4.3)	78.0(5.1)	83.6(2.5)	69.1(4.4)	51.9(2.8)	81.1(2.5)
	WEGL	91.0(3.4)	66.0(2.4)	60.0(2.8)	73.7(1.9)	75.5(1.4)	66.4(2.1)	50.3(1.0)	79.6(0.5)
GNN	PATCHYSAN	91.6(4.6)	58.9(3.7)	55.9(4.5)	75.1(3.3)	76.9(2.3)	62.9(3.9)	45.9(2.5)	73.1(2.7)
	GIN	90.1(4.4)	63.1(3.9)	62.2(3.6)	76.2(2.8)	82.2(0.8)	64.3(3.1)	50.9(1.7)	79.3(1.7)
	DropGIN	89.8(6.2)	62.3(6.8)	65.8(2.7)	76.9(4.3)	81.9(2.5)	66.3(4.5)	51.6(3.2)	80.1(2.8)
	PPGN	90.4(5.6)	65.6(6.0)	66.9(4.3)	77.1(4.0)	82.7(1.8)	67.2(4.1)	51.3(2.8)	81.0(2.1)
	DIFFPOOL	86.1(2.0)	45.0(5.2)	61.0(3.1)	71.7(1.4)	80.9(0.7)	61.1(2.0)	45.8(1.4)	80.8(1.6)
Kernels	FGW - ADJ	82.6(7.2)	55.3(8.0)	72.2(4.0)	72.4(4.7)	74.4(2.1)	70.8(3.6)	48.9(3.9)	80.6(1.5)
	FGW - SP	84.4(7.3)	55.5(7.0)	70.5(6.2)	74.3(3.3)	72.8(1.5)	65.0(4.7)	47.8(3.8)	77.8(2.4)
	WL	87.4(5.4)	56.0(3.9)	69.5(3.2)	74.4(2.6)	85.6(1.2)	67.5(4.0)	48.5(4.2)	78.5(1.7)
	WWL	86.3(7.9)	52.6(6.8)	71.4(5.1)	73.1(1.4)	85.7(0.8)	71.6(3.8)	52.6(3.0)	81.4(2.1)
	Gain with TFGW	**+4.3**	**+4.4**	**+2.9**	**+4.9**	**+2.4**	**+6.7**	**+4.2**	**+2.9**

- Comparison with state of the art approach from GNN and graph kernel methods.
- Systematic and significant gain of performance with GIN+TFGW.
- Gain independent of GNN architecture (GIN or GAT).

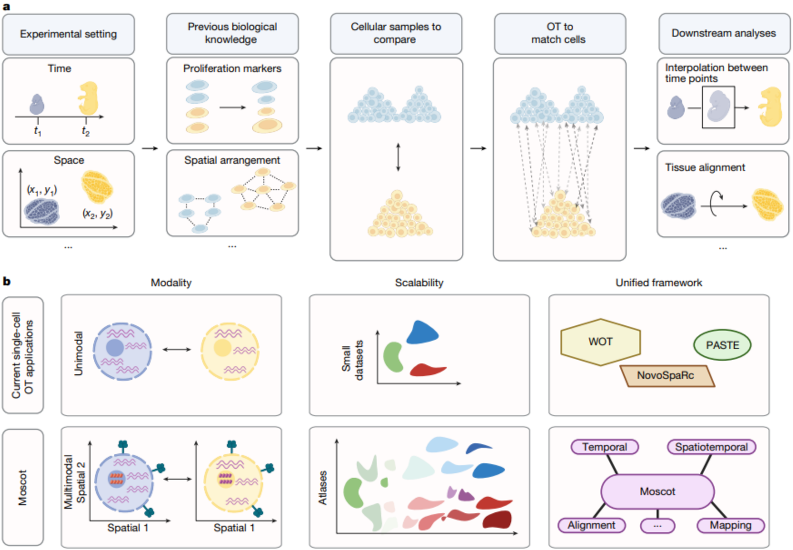
Single cell alignement with OT (SCOT)



Aligning cell population in different modalities [Demetci et al., 2022b]

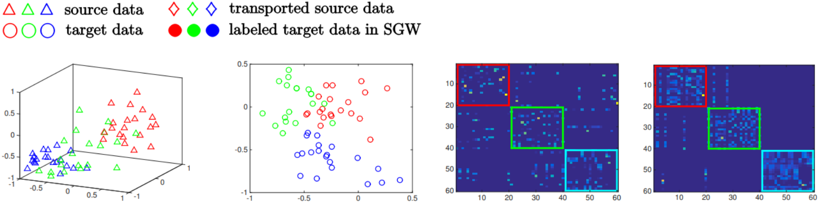
- Population of cells in different modalities (Gene, chromatin).
- Not the same cells because destructive observations.
- Use of Gromov-Wasserstein to recover correspondences.
- Adaptation to cells with different proportions with unbalanced OT [Demetci et al., 2022a, Tran et al., 2023].

Mapping cells through time and space



Moscot: multi-omics single-cell optimal transport [Klein et al., 2025]

Heterogeneous Domain Adaptation with GW



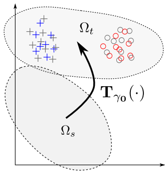
(a) source data (b) target data (c) **T** obtained by EGW (e) **T** obtained by SGW

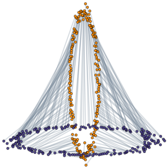
Semi-supervised Heterogeneous Domain Adaptation [Yan et al., 2018]

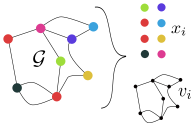
- OT for DA initially proposed by [Courty et al., 2016].
- Use the OT matrix to transfer labels or samples between datasets.
- GW find correspondences across spaces but very noisy.
- Semi-supervised strategy allows very good performances.
- Alternative : Co-optimal transport that find correspondances between the variables and samples simultaneously [Redko et al., 2020].

Outline

Three aspects of optimal transport







Transporting with optimal transport

- Learn to map between distributions.
- Estimate a smooth mapping from discrete distributions.
- Applications in domain adaptation.

Divergence between histograms/empirical distributions

- Use the ground metric to encode complex relations between the bins of histograms for data fitting.
- OT losses are non-parametric divergences between non overlapping distributions.
- Used to train minimal Wasserstein estimators.

Divergence between structured objects and spaces

- Modeling of structured data and graphs as distribution.
- OT losses (Wass. or (F)GW) measure similarity between distributions/objects.
- OT find correspondance across spaces for adaptation.

References i

[Amos et al., 2017] Amos, B., Xu, L., and Kolter, J. Z. (2017).
Input convex neural networks.
In *International Conference on Machine Learning*, pages 146–155. PMLR.

[Arjovsky et al., 2017] Arjovsky, M., Chintala, S., and Bottou, L. (2017).
Wasserstein gan.
arXiv preprint arXiv:1701.07875.

[Barbe et al., 2020] Barbe, A., Sebban, M., Gonçalves, P., Borgnat, P., and Gribonval, R. (2020).
Graph diffusion wasserstein distances.
In *ECML PKDD 2020-European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 1–16.

[Bécigneul et al., 2020] Bécigneul, G., Ganea, O.-E., Chen, B., Barzilay, R., and Jaakkola, T. (2020).
Optimal transport graph neural networks.
arXiv preprint arXiv:2006.04804.

[Benamou et al., 2015] Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).
Iterative Bregman projections for regularized transportation problems.
SISC.

[Bigot et al., 2017] Bigot, J., Gouet, R., Klein, T., López, A., et al. (2017).
Geodesic pca in the wasserstein space by convex pca.
In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 53, pages 1–26. Institut Henri Poincaré.

[Bronstein et al., 2017] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P. (2017).
Geometric deep learning: going beyond euclidean data.
IEEE Signal Processing Magazine, 34(4):18–42.

[Bunne et al., 2022] Bunne, C., Krause, A., and Cuturi, M. (2022).
Supervised training of conditional monge maps.
Advances in Neural Information Processing Systems, 35:6859–6872.

[Bunne et al., 2023] Bunne, C., Stark, S. G., Gut, G., Del Castillo, J. S., Levesque, M., Lehmann, K.-V., Pelkmans, L., Krause, A., and Rätsch, G. (2023).
Learning single-cell perturbation responses using neural optimal transport.
Nature Methods, 20(11):1759–1768.

[Chambon et al., 2018] Chambon, S., Galtier, M. N., and Gramfort, A. (2018).
Domain adaptation with optimal transport improves eeg sleep stage classifiers.
In *2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI)*, pages 1–4. IEEE.

[Courty et al., 2017a] Courty, N., Flamary, R., and Ducoffe, M. (2017a).
Learning wasserstein embeddings.

[Courty et al., 2017b] Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. (2017b).
Joint distribution optimal transportation for domain adaptation.
In *Neural Information Processing Systems (NIPS)*.

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

[Cuturi, 2013] Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transportation.
In *Neural Information Processing Systems (NIPS)*, pages 2292–2300.

[Damodaran et al., 2018] Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018).
Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation.

[Demetci et al., 2022a] Demetci, P., Santorella, R., Chakravarthy, M., Sandstede, B., and Singh, R. (2022a).
Scotv2: Single-cell multiomic alignment with disproportionate cell-type representation.
Journal of Computational Biology, 29(11):1213–1228.

References v

[Demetci et al., 2022b] Demetci, P., Santorella, R., Sandstede, B., Noble, W. S., and Singh, R. (2022b).
Scot: Single-cell multi-omics alignment with optimal transport.
Journal of Computational Biology, 29(1):3–18.

[Fatras et al., 2021] Fatras, K., Bhushan Damodaran, B., Lobry, S., Flamary, R., Tuia, D., and Courty, N. (2021).
Wasserstein adversarial regularization for learning with label noise.
Pattern Analysis and Machine Intelligence, IEEE Transactions on.

[Fatras et al., 2020] Fatras, K., Zine, Y., Flamary, R., Gribonval, R., and Courty, N. (2020).
Learning with minibatch wasserstein : asymptotic and gradient properties.
In *International Conference on Artificial Intelligence and Statistics (AISTAT)*.

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

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

[Flamary et al., 2019] Flamary, R., Lounici, K., and Ferrari, A. (2019).
Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation.
arXiv preprint arXiv:1905.10155.

[Frogner et al., 2015] 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.

[Ganin et al., 2016] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016).
Domain-adversarial training of neural networks.
Journal of Machine Learning Research, 17(59):1–35.

[Gautheron et al., 2017] Gautheron, L., Lartizien, C., and Redko, I. (2017).
Domain adaptation using optimal transport: application to prostate cancer mapping.

[Gayraud et al., 2017] Gayraud, N. T., Rakotomamonjy, A., and Clerc, M. (2017).
Optimal transport applied to transfer learning for p300 detection.
In *BCI 2017-7th Graz Brain-Computer Interface Conference*, page 6.

[Gnassounou et al., 2023] Gnassounou, T., Flamary, R., and Gramfort, A. (2023).
Convolutional monge mapping normalization for learning on biosignals.
In *Neural Information Processing Systems*.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).
Generative adversarial nets.
In *Advances in neural information processing systems*, pages 2672–2680.

[Gulrajani et al., 2017] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017).
Improved training of wasserstein gans.
In *Advances in Neural Information Processing Systems*, pages 5769–5779.

[Klein et al., 2025] Klein, D., Palla, G., Lange, M., Klein, M., Piran, Z., Gander, M., Meng-Papaxanthos, L., Sterr, M., Saber, L., Jing, C., et al. (2025).
Mapping cells through time and space with moscot.
Nature, pages 1–11.

[Korotin et al., 2019] Korotin, A., Egiazarian, V., Asadulaev, A., Safin, A., and Burnaev, E. (2019).
Wasserstein-2 generative networks.
arXiv preprint arXiv:1909.13082.

[Li et al., 2016] Li, P., Rangapuram, S. S., and Slawski, M. (2016).
Methods for sparse and low-rank recovery under simplex constraints.
arXiv preprint arXiv:1605.00507.

[Makkuva et al., 2020] Makkuva, A., Taghvaei, A., Oh, S., and Lee, J. (2020).
Optimal transport mapping via input convex neural networks.
In *International Conference on Machine Learning*, pages 6672–6681. PMLR.

[Maretic et al., 2019] Maretic, H. P., Gheche, M. E., Chierchia, G., and Frossard, P. (2019).
Got: An optimal transport framework for graph comparison.
arXiv preprint arXiv:1906.02085.

[Maretic et al., 2020] Maretic, H. P., Gheche, M. E., Minder, M., Chierchia, G., and Frossard, P. (2020).
Wasserstein-based graph alignment.
arXiv preprint arXiv:2003.06048.

[Mérigot et al., 2020] Mérigot, Q., Delalande, A., and Chazal, F. (2020).
Quantitative stability of optimal transport maps and linearization of the 2-wasserstein space.
In *International Conference on Artificial Intelligence and Statistics*, pages 3186–3196. PMLR.

[Miyato et al., 2018] Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018).
Virtual adversarial training: a regularization method for supervised and semi-supervised learning.
IEEE transactions on pattern analysis and machine intelligence, 41(8):1979–1993.

[Montesuma and Mboula, 2021] Montesuma, E. F. and Mboula, F. M. N. (2021).
Wasserstein barycenter for multi-source domain adaptation.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16785–16793.

[Mroueh, 2019] Mroueh, Y. (2019).
Wasserstein style transfer.
arXiv preprint arXiv:1905.12828.

[Muzellec et al., 2020] Muzellec, B., Josse, J., Boyer, C., and Cuturi, M. (2020).
Missing data imputation using optimal transport.
In *International Conference on Machine Learning*, pages 7130–7140. PMLR.

[Paty et al., 2020] Paty, F.-P., d'Aspremont, A., and Cuturi, M. (2020).
Regularity as regularization: Smooth and strongly convex brenier potentials in optimal transport.
In *International Conference on Artificial Intelligence and Statistics*, pages 1222–1232. PMLR.

References xi

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

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

[Pooladian and Niles-Weed, 2021] Pooladian, A.-A. and Niles-Weed, J. (2021).
Entropic estimation of optimal transport maps.
arXiv preprint arXiv:2109.12004.

[Radford et al., 2015] Radford, A., Metz, L., and Chintala, S. (2015).
Unsupervised representation learning with deep convolutional generative adversarial networks.
arXiv preprint arXiv:1511.06434.

[Redko et al., 2020] Redko, I., Vayer, T., Flamary, R., and Courty, N. (2020).
Co-optimal transport.
In *Neural Information Processing Systems (NeurIPS)*.

[Rivet et al., 2009] Rivet, B., Souloumiac, A., Attina, V., and Gibert, G. (2009).
xdawn algorithm to enhance evoked potentials: application to brain–computer interface.
IEEE Transactions on Biomedical Engineering, 56(8):2035–2043.

[Rolet et al., 2016] 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.

[Sandler and Lindenbaum, 2011] Sandler, R. and Lindenbaum, M. (2011).
Nonnegative matrix factorization with earth mover's distance metric for image analysis.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(8):1590–1602.

[Schiebinger et al., 2019] Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., et al. (2019).
Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming.
Cell, 176(4):928–943.

[Schmitz et al., 2017] Schmitz, M. A., Heitz, M., Bonneel, N., Mboula, F. M. N., Coeurjolly, D., Cuturi, M., Peyré, G., and Starck, J.-L. (2017).
Wasserstein dictionary learning: Optimal transport-based unsupervised non-linear dictionary learning.
arXiv preprint arXiv:1708.01955.

[Seguy et al., 2017] Seguy, V., Bhushan Damodaran, B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. (2017).
Large-scale optimal transport and mapping estimation.

[Seguy and Cuturi, 2015] Seguy, V. and Cuturi, M. (2015).
Principal geodesic analysis for probability measures under the optimal transport metric.
In *Advances in Neural Information Processing Systems*, pages 3312–3320.

[Shen et al., 2018] Shen, J., Qu, Y., Zhang, W., and Yu, Y. (2018).
Wasserstein distance guided representation learning for domain adaptation.
In *AAAI Conference on Artificial Intelligence*.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014).
Very deep convolutional networks for large-scale image recognition.
arXiv preprint arXiv:1409.1556.

[Sun and Saenko, 2016] Sun, B. and Saenko, K. (2016).
Deep CORAL: Correlation Alignment for Deep Domain Adaptation, pages 443–450.
Springer International Publishing, Cham.

[Tran et al., 2023] Tran, Q. H., Janati, H., Courty, N., Flamary, R., Redko, I., Demetci, P., and Singh, R. (2023).
Unbalanced co-optimal transport.
In *Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*.

[Tzeng et al., 2014] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. (2014).
Deep domain confusion: Maximizing for domain invariance.
arXiv preprint arXiv:1412.3474.

[Vincent-Cuaz et al., 2022] Vincent-Cuaz, C., Flamary, R., Corneli, M., Vayer, T., and Courty, N. (2022).
Template based graph neural network with optimal transport distances.
In *Neural Information Processing Systems (NeurIPS)*.

[Vincent-Cuaz et al., 2021] Vincent-Cuaz, C., Vayer, T., Flamary, R., Corneli, M., and Courty, N. (2021).
Online graph dictionary learning.
In *International Conference on Machine Learning (ICML)*.

[Yan et al., 2018] Yan, Y., Li, W., Wu, H., Min, H., Tan, M., and Wu, Q. (2018).
Semi-supervised optimal transport for heterogeneous domain adaptation.
In *IJCAI*, pages 2969–2975.

[Yang et al., 2018] Yang, Q., Yan, P., Zhang, Y., Yu, H., Shi, Y., Mou, X., Kalra, M. K., Zhang, Y., Sun, L., and Wang, G. (2018).
Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss.
IEEE transactions on medical imaging, 37(6):1348–1357.

[Zen et al., 2014] Zen, G., Ricci, E., and Sebe, N. (2014).
Simultaneous ground metric learning and matrix factorization with earth mover's distance.
In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 3690–3695.

[Zhao et al., 2016] Zhao, J., Mathieu, M., and LeCun, Y. (2016).
Energy-based generative adversarial network.
arXiv preprint arXiv:1609.03126.