Flowing Datasets with Wasserstein over Wasserstein Gradient Flows

Clément Bonet*[1], Christophe Vauthier*[2], Anna Korba[1]

[1]ENSAE, CREST, Institut Polytechnique de Paris
[2]Université Paris-Saclay, Laboratoire de Mathématique d'Orsay

ICML 2025

Motivations

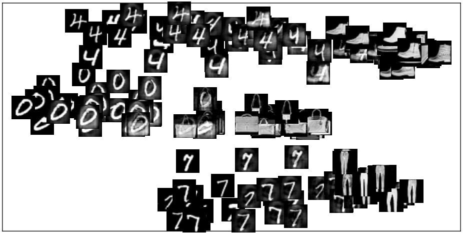
Labeled dataset: $\mathcal{D} = \big((x_i, y_i)\big)_{i=1}^{n}$, $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$

Typically: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{1, \ldots, C\}$, C: number of classes

Motivations

Labeled dataset: $\mathcal{D} = \big((x_i, y_i)\big)_{i=1}^{n}$, $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$
Typically: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{1, \ldots, C\}$, C: number of classes

Goal: Generate samples from $\mathcal{D}$ respecting the structure of the dataset



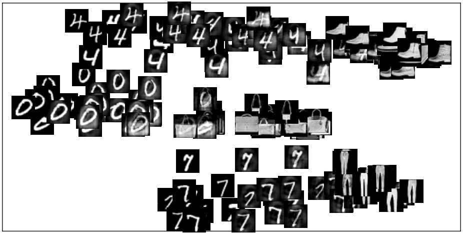
Flowing MNIST towards Fashion MNIST

Motivations

Labeled dataset: $\mathcal{D} = \left((x_i, y_i)\right)_{i=1}^n$, $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$

Typically: $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{1, \ldots, C\}$, C: number of classes

Goal: Generate samples from $\mathcal{D}$ respecting the structure of the dataset



Flowing MNIST towards Fashion MNIST

Applications:

- Domain adaptation (Courty et al., 2016)
- Transfer learning (Alvarez-Melis and Fusi, 2021; Hua et al., 2023)
- Dataset distillation (Wang et al., 2018)

OTDD (Alvarez-Melis and Fusi, 2020)

- $\mathcal{D}_1$ (MNIST): $\mu_1 = \frac{1}{m}\sum_{i=1}^m \delta_{(x_i^1, y_i^1)} \in \mathcal{P}(\mathbb{R}^d \times \{1, \ldots, C\})$,
  $\mathcal{D}_2$ (Fashion MNIST): $\mu_2 = \frac{1}{m}\sum_{j=1}^m \delta_{(x_j^2, y_j^2)} \in \mathcal{P}(\mathbb{R}^d \times \{1, \ldots, C\})$
  C: number of classes, n: number of sample in each class, $m = nC$

Question: how to compare datasets $\mathcal{D}_1$ and $\mathcal{D}_2$?

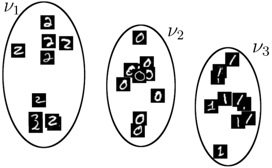
OTDD (Alvarez-Melis and Fusi, 2020)

- $\mathcal{D}_1$ (MNIST): $\mu_1 = \frac{1}{m}\sum_{i=1}^{m} \delta_{(x_i^1, y_i^1)} \in \mathcal{P}(\mathbb{R}^d \times \{1, \ldots, C\})$,
  $\mathcal{D}_2$ (Fashion MNIST): $\mu_2 = \frac{1}{m}\sum_{j=1}^{m} \delta_{(x_j^2, y_j^2)} \in \mathcal{P}(\mathbb{R}^d \times \{1, \ldots, C\})$
  C: number of classes, n: number of sample in each class, $m = nC$

Question: how to compare datasets $\mathcal{D}_1$ and $\mathcal{D}_2$?

Solution of Alvarez-Melis and Fusi (2020):

- Embed a label (a class) in $\mathcal{P}(\mathbb{R}^d)$ as $c \mapsto \nu_c = \hat{P}(\cdot|y = c)$



$\rightarrow \mathcal{D}_k : \mu_k = \frac{1}{m}\sum_{i=1}^{m} \delta_{(x_i^k, \nu_{y_i^k}^k)} \in \mathcal{P}\big(\mathbb{R}^d \times \mathcal{P}(\mathbb{R}^d)\big)$

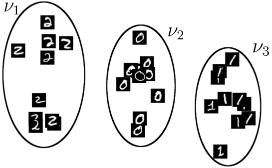
OTDD (Alvarez-Melis and Fusi, 2020)

- $\mathcal{D}_1$ (MNIST): $\mu_1 = \frac{1}{m}\sum_{i=1}^{m} \delta_{(x_i^1, y_i^1)} \in \mathcal{P}(\mathbb{R}^d \times \{1, \ldots, C\})$,
  $\mathcal{D}_2$ (Fashion MNIST): $\mu_2 = \frac{1}{m}\sum_{j=1}^{m} \delta_{(x_j^2, y_j^2)} \in \mathcal{P}(\mathbb{R}^d \times \{1, \ldots, C\})$
  C: number of classes, n: number of sample in each class, $m = nC$

Question: how to compare datasets $\mathcal{D}_1$ and $\mathcal{D}_2$?

Solution of Alvarez-Melis and Fusi (2020):

- Embed a label (a class) in $\mathcal{P}(\mathbb{R}^d)$ as $c \mapsto \nu_c = \hat{P}(\cdot | y = c)$



$\rightarrow \mathcal{D}_k : \mu_k = \frac{1}{m}\sum_{i=1}^{m} \delta_{(x_i^k, \nu_{y_i^k}^k)} \in \mathcal{P}\big(\mathbb{R}^d \times \mathcal{P}(\mathbb{R}^d)\big)$

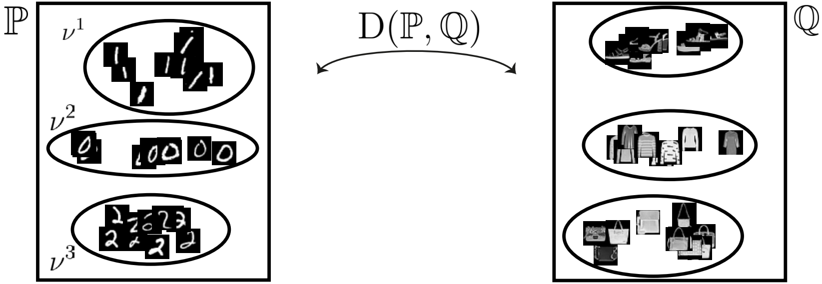
- Cost: $d\big((x,y),(x',y')\big)^2 = \|x - x'\|_2^2 + \mathrm{W}_2^2(\nu_y, \nu_{y'})$
- Optimal transport distance: approximated in $O\big(n^2 C^2 \log(nC)\big)$

$$\mathrm{OTDD}(\mu_1, \mu_2) = \inf_{\gamma \in \Pi(\mu_1, \mu_2)} \int d\big((x,y),(x',y')\big)^2 \, \mathrm{d}\gamma\big((x,y),(x',y')\big).$$

Contributions

- Model datasets as $\mathbb{P} = \frac{1}{C}\sum_{c=1}^{C} \delta_{\nu^c} \in \mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big)$ where $\nu^c = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i^c}$
- Flow a dataset $\mathbb{P}$ towards $\mathbb{Q}$ by minimizing a discrepancy D on $\mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big)$

$\rightarrow$ **minimization problem** on $\mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big)$



Example

$\mathrm{D}(\mathbb{P}, \mathbb{Q}) = \mathrm{MMD}_K^2(\mathbb{P}, \mathbb{Q})$ with $K : \mathcal{P}(\mathbb{R}^d) \times \mathcal{P}(\mathbb{R}^d) \rightarrow \mathbb{R}$

Wasserstein over Wasserstein (WoW) Space

Wasserstein over Wasserstein distance: Let $\mathbb{P}, \mathbb{Q} \in \mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big)$,

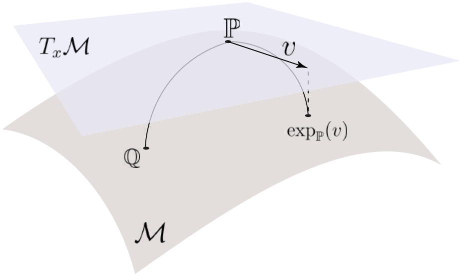
$$\mathrm{W}^2_{\mathrm{W}_2}(\mathbb{P}, \mathbb{Q}) = \inf_{\Gamma \in \Pi(\mathbb{P}, \mathbb{Q})} \int \mathrm{W}_2^2(\mu, \nu)\ \mathrm{d}\Gamma(\mu, \nu)$$

WoW space $\big(\mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big), \mathrm{W}_{\mathrm{W}_2}\big)$: Riemannian structure

$\rightarrow$ Geodesics between $\mathbb{P}, \mathbb{Q} \in \mathcal{M} = \mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big)$

$\rightarrow$ Exponential map: $\forall \mathbb{P} \in \mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big)$, $\exp_{\mathbb{P}} : T_{\mathbb{P}}\mathcal{M} \rightarrow \mathcal{M}$

$\rightarrow$ Gradient of $\mathbb{F} : \mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big) \rightarrow \mathbb{R}$: $\nabla_{\mathrm{W}_{\mathrm{W}_2}} \mathbb{F}(\mathbb{P}) \in T_{\mathbb{P}}\mathcal{M}$



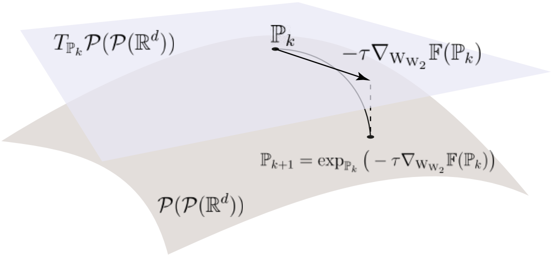
Wasserstein over Wasserstein (WoW) Space

Wasserstein over Wasserstein distance: Let $\mathbb{P}, \mathbb{Q} \in \mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big)$,

$$\mathrm{W}_{\mathrm{W}_2}^2(\mathbb{P}, \mathbb{Q}) = \inf_{\Gamma \in \Pi(\mathbb{P},\mathbb{Q})} \int \mathrm{W}_2^2(\mu, \nu)\ \mathrm{d}\Gamma(\mu, \nu)$$

Minimization of $\mathbb{F} : \mathcal{P}\big(\mathcal{P}(\mathbb{R}^d)\big) \to \mathbb{R}$
$\to$ **WoW Gradient Descent**

$$\forall k \geq 0,\ \mathbb{P}_{k+1} = \exp_{\mathbb{P}_k}\big(-\tau \nabla_{\mathrm{W}_{\mathrm{W}_2}} \mathbb{F}(\mathbb{P}_k)\big)$$



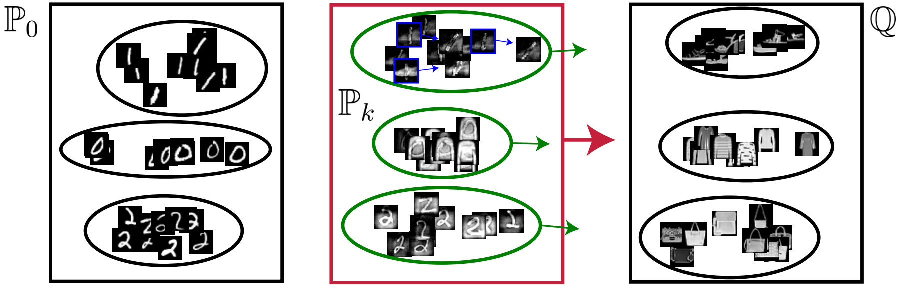
WoW Gradient Flow

Goal: minimize $\mathbb{F}(\mathbb{P}) = \mathrm{D}(\mathbb{P}, \mathbb{Q})$

In practice: For $\mathbb{P}_k = \frac{1}{C}\sum_{c=1}^{C} \delta_{\mu_k^{c,n}}$ with $\mu_k^{c,n} = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_{i,k}^c}$:

$$\forall k \geq 0,\ \text{particle (image) } i,\ \text{class } c,\ x_{i,k+1}^c = x_{i,k}^c - \tau \nabla_{\mathrm{W}_{\mathrm{W}_2}} \mathbb{F}(\mathbb{P}_k)(\mu_k^{c,n})(x_{i,k}^c).$$

$\mathbb{P}_k$: inter-class interaction, $\mu_k^{c,n}$: intra-class interaction, $x_{i,k}^c$ image



Synthetic Data

$$\mathbb{F}(\mathbb{P}) = \frac{1}{2}\mathrm{MMD}_K^2(\mathbb{P},\mathbb{Q}) = \frac{1}{2}\iint K(\mu,\nu)\ \mathrm{d}(\mathbb{P}-\mathbb{Q})(\mu)\mathrm{d}(\mathbb{P}-\mathbb{Q})(\nu)$$

- WoW gradient computed in **closed-form** or using **auto-differentiation**
- Kernel K based on the **Sliced-Wasserstein** distance

Sliced-Wasserstein distance (Rabin et al., 2011; Bonneel et al., 2015):

$$\mathrm{SW}_2^2(\mu,\nu) = \int_{S^{d-1}} \mathrm{W}_2^2(P^\theta_\#\mu, P^\theta_\#\nu)\mathrm{d}\sigma(\theta) \approx \frac{1}{L}\sum_{\ell=1}^{L} \mathrm{W}_2^2(P^{\theta_\ell}_\#\mu, P^{\theta_\ell}_\#\nu),$$

with $S^{d-1} = \{\theta \in \mathbb{R}^d,\ \|\theta\|_2 = 1\}$, $P^\theta(x) = \langle x, \theta\rangle$, $\theta_1, \dots, \theta_L \sim \sigma = \mathrm{Unif}(S^{d-1})$.

- Gaussian SW kernel: $K(\mu,\nu) = e^{-\mathrm{SW}_2^2(\mu,\nu)/h}$ (Kolouri et al., 2016)
- Riesz SW kernel: $K(\mu,\nu) = -\mathrm{SW}_2(\mu,\nu)$

Complexity: $O(C^2 L n \log n)$

C: number of classes of $\mathbb{P}$, n: number of samples in each class,

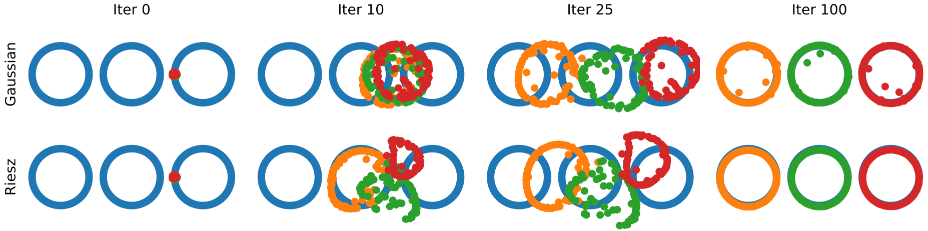
L: number of directions to approximate SW

Synthetic Data

$$\mathbb{F}(\mathbb{P}) = \frac{1}{2}\mathrm{MMD}_K^2(\mathbb{P},\mathbb{Q}) = \frac{1}{2}\iint K(\mu,\nu)\ \mathrm{d}(\mathbb{P}-\mathbb{Q})(\mu)\mathrm{d}(\mathbb{P}-\mathbb{Q})(\nu)$$

- WoW gradient computed in **closed-form** or using **auto-differentiation**
- Kernel K based on the **Sliced-Wasserstein** distance

Goal: $\min_{\mathbb{P}}\ \mathbb{F}(\mathbb{P}) = \frac{1}{2}\mathrm{MMD}_K^2(\mathbb{P},\mathbb{Q})$, where $\mathbb{Q} = \frac{1}{3}\sum_{c=1}^{3}\delta_{\nu^{c,n}}$, $\nu^{c,n}$ ring.

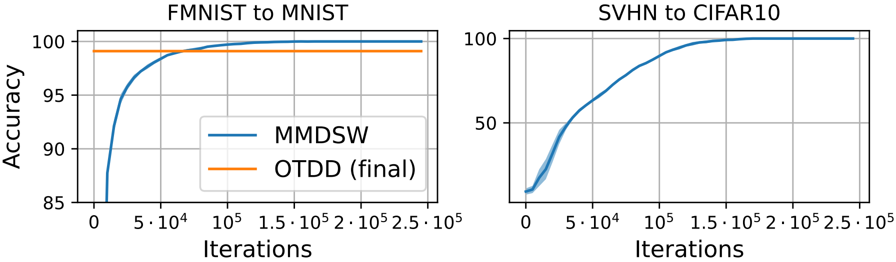


$\rightarrow$ The flow is **preserving the class (ring) structure**.

Domain Adaptation

Setting:

1. Pretrain a classifier on $\mathbb{Q}$ =MNIST (**Left**) or $\mathbb{Q}$ =CIFAR10 (**Right**)
2. Flow starting from $\mathbb{P}_0$ =Fashion MNIST (**Left**) or from $\mathbb{P}_0$ =SVHN (**Right**) by minimizing $\mathbb{F}(\mathbb{P}) = \frac{1}{2}\mathrm{MMD}_K^2(\mathbb{P}, \mathbb{Q})$ with $K(\mu, \nu) = -\mathrm{SW}_2(\mu, \nu)$
3. Measure accuracy on $\mathbb{P}_t$ (flowed data)

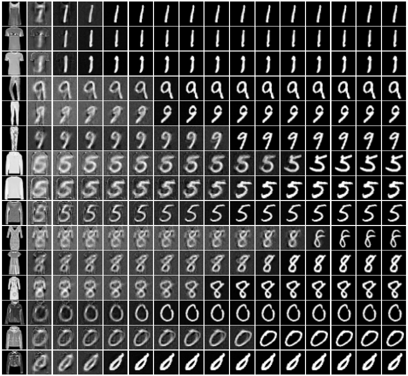


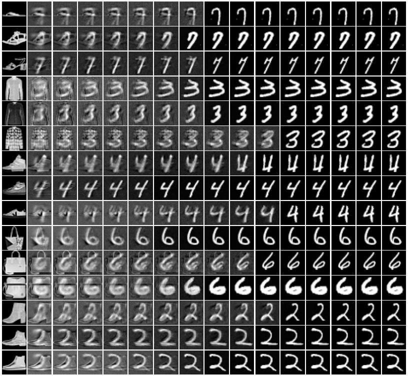
→ reach 100% accuracy: flowed data are well classified

Domain Adaptation

Setting:

1. Pretrain a classifier on $\mathbb{Q}$ =MNIST (**Left**) or $\mathbb{Q}$ =CIFAR10 (**Right**)
2. Flow starting from $\mathbb{P}_0$ =Fashion MNIST (**Left**) or from $\mathbb{P}_0$ =SVHN (**Right**) by minimizing $\mathbb{F}(\mathbb{P}) = \frac{1}{2}\mathrm{MMD}_K^2(\mathbb{P}, \mathbb{Q})$ with $K(\mu, \nu) = -\mathrm{SW}_2(\mu, \nu)$
3. Measure accuracy on $\mathbb{P}_t$ (flowed data)





$\rightarrow$ **Data from a same class are flowed towards a same class**

Dataset Distillation

Dataset distillation: synthesize a big dataset $\mathbb{Q} = \frac{1}{C}\sum_{c=1}^C \delta_{\nu^{c,n}}$ with a small dataset $\mathbb{P} = \frac{1}{C}\sum_{c=1}^C \delta_{\mu^{c,p}}$, p **small**

- Distribution Matching (DM) (Zhao and Bilen, 2023): For $k(x,y) = \langle x, y\rangle$,

$$\mathcal{F}\big((\mu^c)_c\big) = \sum_{c=1}^C \mathrm{MMD}_k^2(\mu^c, \nu^c)\text{: distillation per class}$$

- Our method (**MMDSW**): For $K(\mu,\nu) = -\mathrm{SW}_2(\mu,\nu)$, $\mathbb{P} = \frac{1}{C}\sum_{c=1}^C \delta_{\mu^c}$,

$$\tilde{\mathbb{F}}(\mathbb{P}) = \mathrm{MMD}_K^2(\mathbb{P},\mathbb{Q})\text{: } \textbf{distillation on the whole dataset}$$

A: with random augmentation, E: with random embeddings

Dataset	p	no A, no E		A		E		A and E		Baselines	
		DM	MMDSW	DM	MMDSW	DM	MMDSW	DM	MMDSW	Random	Full data
MNIST	1	$61.1_{\pm 6.5}$	$\mathbf{66.5}_{\pm 5.5}$	-	$\mathbf{66.8}_{\pm 5.3}$	$\mathbf{87.8}_{\pm 0.6}$	$60.3_{\pm 3.4}$	$\mathbf{87.7}_{\pm 0.5}$	$60.9_{\pm 3.3}$	$55.8_{\pm 2.0}$	
	10	$88.2_{\pm 2.8}$	$\mathbf{93.2}_{\pm 0.7}$	$88.7_{\pm 3.3}$	$\mathbf{93.8}_{\pm 0.7}$	$\mathbf{97.0}_{\pm 0.1}$	$96.4_{\pm 0.2}$	$\mathbf{97.0}_{\pm 0.1}$	$96.4_{\pm 0.3}$	$92.2_{\pm 1.1}$	99.4
	50	$95.9_{\pm 0.9}$	$97.0_{\pm 0.2}$	$95.3_{\pm 1.4}$	$97.5_{\pm 0.1}$	$\mathbf{98.4}_{\pm 0.1}$	$\mathbf{98.4}_{\pm 0.1}$	$\mathbf{98.4}_{\pm 0.1}$	$\mathbf{98.4}_{\pm 0.1}$	$97.6_{\pm 0.2}$	
FMNIST	1	$54.4_{\pm 3.2}$	$\mathbf{60.0}_{\pm 4.1}$	-	$\mathbf{60.6}_{\pm 3.6}$	$58.7_{\pm 0.4}$	$\mathbf{60.9}_{\pm 2.6}$	$58.7_{\pm 0.5}$	$\mathbf{60.8}_{\pm 2.2}$	$49.0_{\pm 7.5}$	
	10	$74.6_{\pm 1.0}$	$\mathbf{76.7}_{\pm 1.0}$	$74.7_{\pm 0.8}$	$\mathbf{76.6}_{\pm 1.1}$	$\mathbf{81.2}_{\pm 2.3}$	$78.0_{\pm 0.9}$	$\mathbf{82.5}_{\pm 0.3}$	$78.9_{\pm 1.2}$	$75.3_{\pm 0.7}$	92.4
	50	$81.3_{\pm 0.5}$	$\mathbf{84.2}_{\pm 0.1}$	$81.4_{\pm 1.0}$	$\mathbf{85.0}_{\pm 0.2}$	$\mathbf{87.6}_{\pm 0.2}$	$\mathbf{87.6}_{\pm 0.2}$	$87.5_{\pm 0.1}$	$\mathbf{87.6}_{\pm 0.2}$	$83.2_{\pm 0.2}$	

$\rightarrow$ Comparable performances between DM and MMDSW

Transfer Learning

Goal: augment small dataset $\mathbb{Q} = \frac{1}{C}\sum_{c=1}^{C} \delta_{\nu^{c,k}}$ with k small

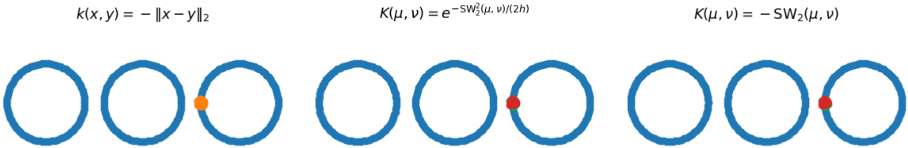
1. Flow a large source dataset $\mathbb{P} =$ MNIST the small target one $\mathbb{Q} =$KMNIST
2. Concatenate the true samples of $\mathbb{Q}$ and the synthetic ones (the source flowed) and train the classifier

Dataset	k	Train on $\mathbb{Q}$	**MMDSW** (ours)	OTDD	(Hua et al., 2023)
MNIST to KMNIST	1	$18.4_{\pm 3.1}$	$\mathbf{20.9}_{\pm 2.0}$	$18.8_{\pm 2.1}$	$19.4_{\pm 1.9}$
	5	$25.9_{\pm 4.0}$	$37.4_{\pm 2.2}$	$31.3_{\pm 1.4}$	$\mathbf{39.0}_{\pm 1.0}$
	10	$30.9_{\pm 4.6}$	$\mathbf{44.7}_{\pm 1.8}$	$34.1_{\pm 0.9}$	$44.1_{\pm 1.2}$
	100	$60.1_{\pm 1.1}$	$\mathbf{66.8}_{\pm 0.8}$	$66.3_{\pm 0.9}$	$62.4_{\pm 1.2}$

$\rightarrow$ **Better performances with smaller computational complexity**

Conclusion

Thank you!



See you at the poster session! **East Exhibition Hall A-B #E-1300**

References I

David Alvarez-Melis and Nicolo Fusi. Geometric Dataset Distances via Optimal Transport. *Advances in Neural Information Processing Systems*, 33: 21428–21439, 2020.

David Alvarez-Melis and Nicolò Fusi. Dataset Dynamics via Gradient Flows in Probability Space. In *International conference on machine learning*, pages 219–230. PMLR, 2021.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2016.

Xinru Hua, Truyen Nguyen, Tam Le, Jose Blanchet, and Viet Anh Nguyen. Dynamic Flows on Curved Space Generated by Labeled Data. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3803–3811. International Joint Conferences on Artificial Intelligence Organization, 8 2023. Main Track.

References II

Soheil Kolouri, Yang Zou, and Gustavo K Rohde. Sliced Wasserstein Kernels for Probability Distributions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5267, 2016.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International conference on scale space and variational methods in computer vision*, pages 435–446. Springer, 2011.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset Distillation. *arXiv preprint arXiv:1811.10959*, 2018.

Bo Zhao and Hakan Bilen. Dataset Condensation with Distribution Matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023.