Dataset Dynamics via Gradient Flows in Probability Space

David Alvarez-Melis
Nicolò Fusi
Microsoft

ICML International Conference On Machine Learning

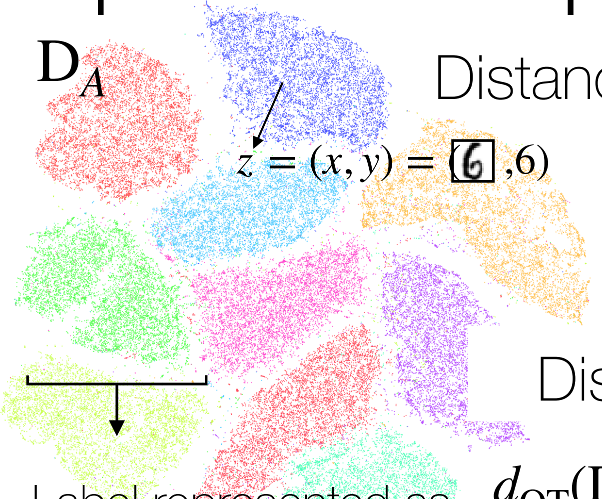
Gradient Flows…

- GF: curve $x(t)$ of steepest descent on functional $F : \mathcal{X} \to \mathbb{R}$
- In Euclidean space $\mathcal{X}$:
$$x'(t) = -\nabla F(x(t))$$
- In Probability space $\mathcal{P}(\mathcal{X})$:
$$\partial_t \rho_t = \nabla \cdot \left(\rho_t \nabla \frac{\delta F}{\delta \rho}\right)$$

Motivation & Summary

- Dataset **transformation**: ubiquitous in ML, from augmentation to generation
- Need often arises because available (generic) data $\neq$ needed (task-specific) data
- Here: general, principled, efficient **labeled** dataset **transformation by optimization**
- Solved using **gradient flows**: guarantees, efficient computation, yields full path
- Vision: **data-centric learning** paradigm, complementary to model-centric one

Optimal Transport Dataset Distance



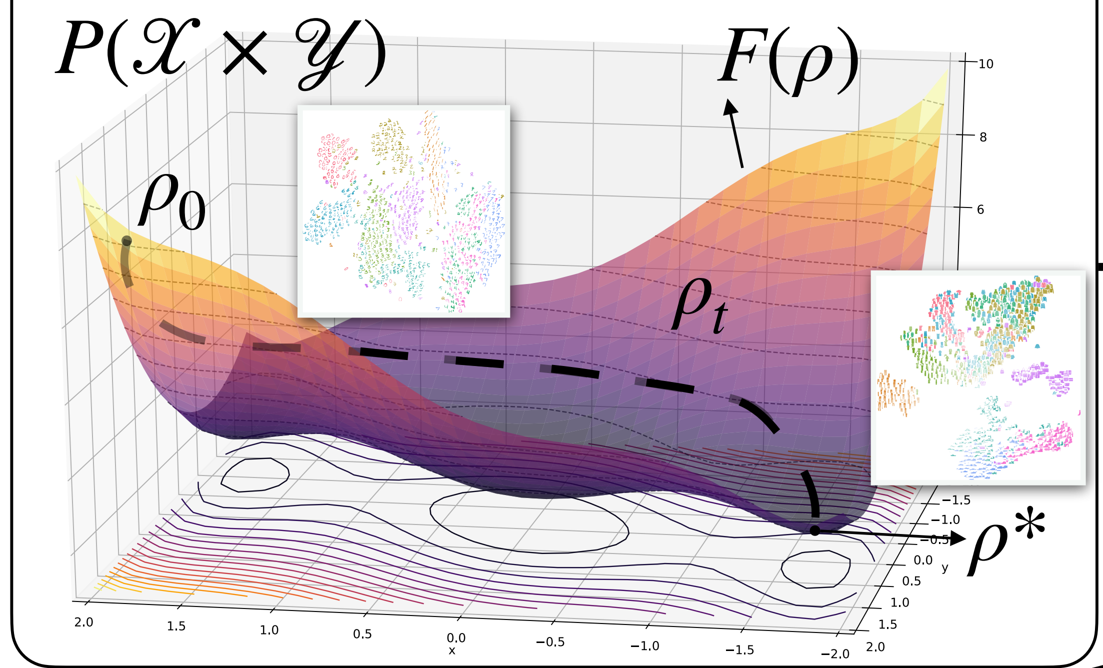
Label represented as distribution over features $v_y = \mathcal{N}(\mu_y, \Sigma_y)$

Distance between **feature/label pairs**:
$$d(z, z') = \Big(d(x, x')^p + \underbrace{\mathrm{W}_p^p(v_y, v_{y'})}_{\text{Wasserstein distance}}\Big)^{1/p}$$

Distance between **datasets**:
$$d_{\mathrm{OT}}(\mathrm{D}_A, \mathrm{D}_B) = \min_{\pi \in \Pi(\alpha,\beta)} \int_{\mathcal{Z}\times\mathcal{Z}} d(z, z')^p \, \mathrm{d}\pi(z, z')$$

OT distance: ≈ min-cost matching

… in dataset space



Dataset Transformation using Gradient Flows

- Model $F(z)$ via well-behaved functionals:

$\mathcal{F}(\rho) = \int f(\rho(z))\mathrm{d}z$ — internal energy

$\mathcal{V}(\rho) = \int V(z)\mathrm{d}\rho$ — potential

$\mathcal{W}(\rho) = \frac{1}{2}\iint W(z - z')\mathrm{d}\rho(z)\mathrm{d}\rho(z')$ — interaction

$\mathcal{T}(\rho) = \mathrm{OTDD}(\rho, \beta)$ — distance

→ have pop. convergence guarantees, simple 'derivative', tractable

can model various useful objectives on datasets

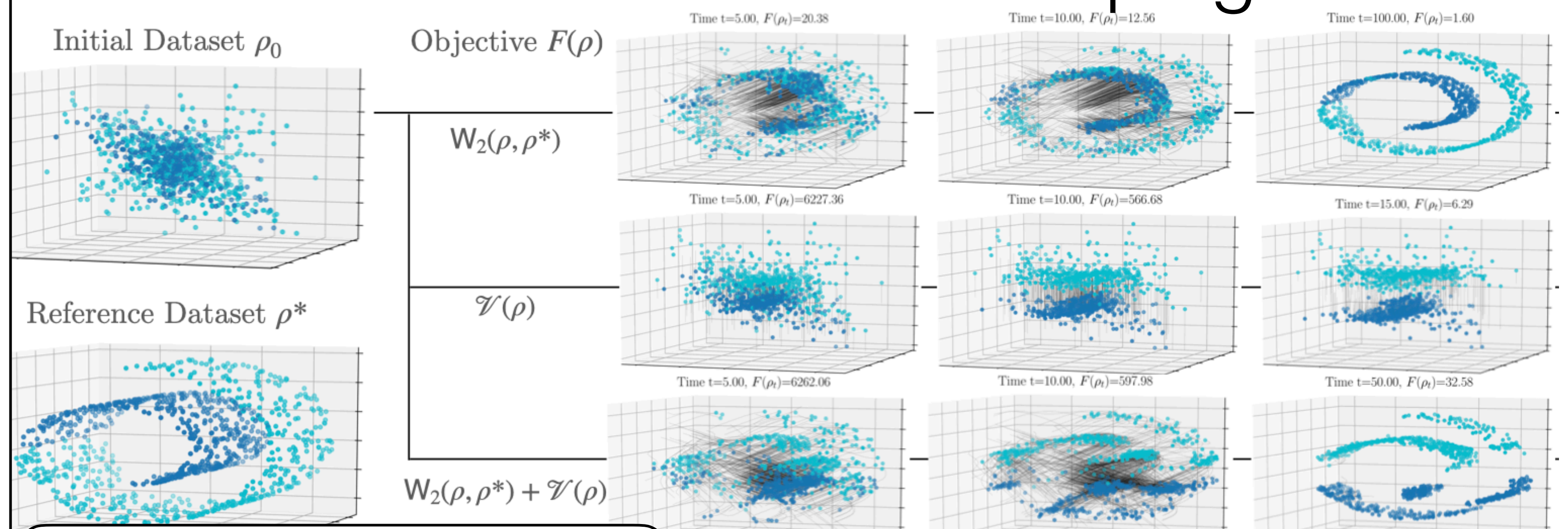
Objective: $\min_{\rho \in \mathcal{P}(\mathcal{X}\times\mathcal{Y})} F(\rho)$

Flow: $\partial_t \rho_t(z) = \nabla \cdot \left(\rho_t(z) \nabla \frac{\delta F}{\delta \rho}(z)\right)$

Practical Implementation

- Flow discretized in time (Euler) & space (particles):
$$z_{t+1}^{(i)} = z_t^{(i)} - \gamma \nabla_z F(z_t^{(i)}), \quad i \in \{i, \ldots, n\}$$
- Distribution is approximated as $\hat{\rho}_t = \sum p_i \delta_{z_t^{(i)}}$
- Implemented using automatic differentiation
- Challenge for $F =$ OTDD: how to update labels, we propose three types of update schemes
- Unlabeled data? Semi- and un-supervised flows

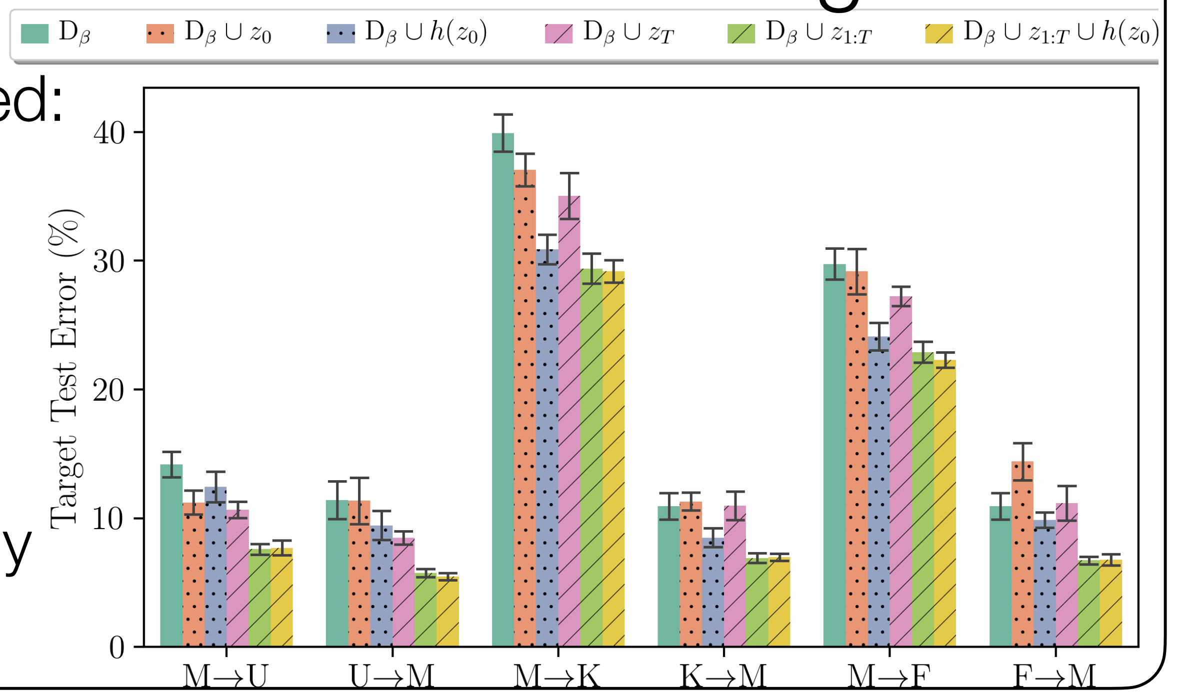
Flows for Dataset Shaping



Want: ρ^* but linearly separable

Flows for Transfer Learning

- Semi-supervised:
  1. run flow with labeled data
  2. fit parametrized model of flow
- Flowed data helps, especially full trajectories!



Flows for Model Re-Purposing

- ResNet trained on CIFAR10, **frozen**
- Target dataset: Camelyon10
- Flow: CAM→CIFAR
- **High acc. on flowed data!**