Google DeepMind

Differentiable and Sparse Top-k: a Convex Analysis Perspective

Mathieu Blondel

July 28th, ICML 2023

Michaël Sander

Joan Puigcerver

Josip Djolonga

Gabriel Peyré

Mathieu Blondel

Motivation for the research

- The top-k operator is increasingly used as a **building block** in neural networks (top-k classification, mixtures of expert, weight pruning)
- However, it is a **discontinuous** operation, making it difficult to use in end-to-end trainable networks
- A crucial property of the top-k is its **sparsity** but many existing differentiable top-k relaxations are **dense**
- **Smooth optimization** is known to enjoy **faster** convergence rates
- However, sparsity is crucial in certain applications as a **selection mechanism**: mixtures of experts, weight pruning

Related work

A large body of work on relaxations of sorting, ranking and top-k...

- Using **unimodal row-stochastic matrices** (Grover et al, 2019; Prillo and Eisenschlos, 2020)
- Using **optimal transport** (Cuturi et al, 2019)
- Using the **permutahedron** (Blondel et al, 2020)
- Using **perturbations** (Berthet et al, 2020)
- Using **sorting networks** (Petersen et al, 2021)

Contributions

- A **general** top-k framework, including top-k **in magnitude**
- Differentiable **and** sparse relaxations thanks to p-norm regularization
- Reduction to **isotonic optimization**, for computation and differentiation
- **GPU/TPU-friendly** algorithm based on Dykstra's algorithm
- Applications to top-k classification, mixtures of experts, weight pruning

1 Top-k mask

Top-k mask operator

Bit-encoding of the top-k indices ("k-hot encoding")

$$[\mathbf{topkmask}(\boldsymbol{x})]_i := \begin{cases} 1, & \text{if } [\mathbf{rank}(\boldsymbol{x})]_i \leq k \\ 0, & \text{otherwise.} \end{cases} \in \{0,1\}^n$$

$$\begin{aligned} \boldsymbol{x} &= (1.7, 3.2, -2.4) \\ \mathbf{top1mask}(\boldsymbol{x}) &= (0,1,0) \\ \mathbf{top2mask}(\boldsymbol{x}) &= (1,1,0) \\ \mathbf{top3mask}(\boldsymbol{x}) &= (1,1,1) \end{aligned}$$

Discontinuous, piecewise constant with null derivatives

Top-k operator

Sparse vector containing the top-k values

$$\mathbf{topk}(\boldsymbol{x}) \coloneqq \boldsymbol{x} \cdot \mathbf{topkmask}(\boldsymbol{x}) \in \mathbb{R}^n$$

$$\begin{aligned}
\boldsymbol{x} &= (1.7, 3.2, -2.4) \\
\mathbf{top1}(\boldsymbol{x}) &= (0.0, 3.2, 0) \\
\mathbf{top2}(\boldsymbol{x}) &= (1.7, 3.2, 0) \\
\mathbf{top3}(\boldsymbol{x}) &= (1.7, 3.2, -2.4)
\end{aligned}$$

Discontinuous, piecewise affine with constant derivatives

Regularized top-k mask: overview of the approach

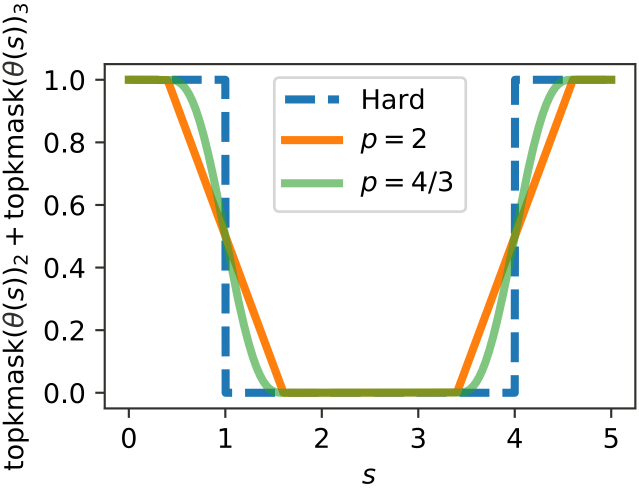
- Rewrite top-k mask as a linear program solution

$$\mathbf{topkmask}(\boldsymbol{x}) = \boldsymbol{y}(\boldsymbol{x}) := \underset{\boldsymbol{y} \in \mathcal{C}}{\mathbf{argmax}} \langle \boldsymbol{x}, \boldsymbol{y} \rangle$$

- Add regularization R

$$\mathbf{topkmask}_R(\boldsymbol{x}) = \boldsymbol{y}_R(\boldsymbol{x}) := \underset{\boldsymbol{y} \in \mathcal{C}}{\mathbf{argmax}} \langle \boldsymbol{x}, \boldsymbol{y} \rangle - R(\boldsymbol{y})$$

- Use a reduction to isotonic optimization to easily compute and differentiate $\mathbf{topkmask}_R(\boldsymbol{x})$



$$\theta(s) = (3, 1, -1 + s, s) \in \mathbb{R}^4$$
$$k = 2$$

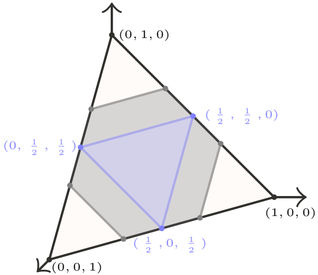
Top-k mask as a linear program

- With $\mathcal{C} = \{\boldsymbol{y} \in \mathbb{R}^n \colon \boldsymbol{y} \in [0,1]^n, \boldsymbol{y}^\top \mathbf{1} = k\}$, we get

$$\textbf{topkmask}(\boldsymbol{x}) = \boldsymbol{y}(\boldsymbol{x}) = \underset{\boldsymbol{y} \in \mathcal{C}}{\textbf{argmax}} \langle \boldsymbol{x}, \boldsymbol{y} \rangle$$

- The vertices of $\mathcal{C}$ are all possible bit encodings of cardinality k
- Relation with the **capped probability simplex**

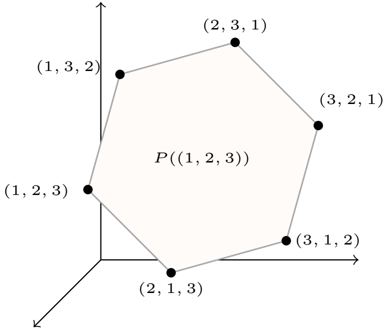
$$\mathcal{C}/k = \left\{ \boldsymbol{y} \in \mathbb{R}^n \colon \boldsymbol{y} \in [0, 1/k]^n, \boldsymbol{y}^\top \mathbf{1} = 1 \right\}$$



Relation with the permutahedron

- The convex hull of all permutations of $\boldsymbol{w}$

$$P(\boldsymbol{w}) := \mathbf{conv}(\{(w_{\sigma_1}, \ldots, w_{\sigma_n}) \colon \sigma \in \Sigma\})$$



- With $\boldsymbol{w} = \mathbf{1}_k := (\underbrace{1, \ldots, 1}_{k}, \underbrace{0, \ldots, 0}_{n-k})$, we get

$$P(\boldsymbol{w}) = \mathcal{C} = \{\boldsymbol{y} \in \mathbb{R}^n \colon \boldsymbol{y} \in [0,1]^n, \boldsymbol{y}^\top \mathbf{1} = k\}$$

Top-k mask: value function and its conjugate

- Value function: support function of $\mathcal{C}$

$$f(\boldsymbol{x}) := \max_{\boldsymbol{y}\in\mathcal{C}} \langle \boldsymbol{x}, \boldsymbol{y} \rangle = \textbf{topksum}(\boldsymbol{x}) := \sum_{i=1}^{k} x_{\sigma_i} = \langle \boldsymbol{x}_\sigma, \mathbf{1}_k \rangle$$

where $\sigma = \textbf{argsort}(\boldsymbol{x}) \iff x_{\sigma_1} \geq \cdots \geq x_{\sigma_n}$ and $\boldsymbol{x}_\sigma := (x_{\sigma_1}, \ldots, x_{\sigma_n})$

- Conjugate: indicator function of $\mathcal{C}$

$$f^*(\boldsymbol{y}) := \sup_{\boldsymbol{x}\in\mathbb{R}^n} \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}) = \delta_{\mathcal{C}}(\boldsymbol{y}) := \begin{cases} 0, & \text{if } \boldsymbol{y} \in \mathcal{C} \\ \infty, & \text{if } \boldsymbol{y} \notin \mathcal{C} \end{cases}$$

Regularized version

- The regularized version

$$\mathbf{topkmask}_R(\boldsymbol{x}) = \boldsymbol{y}_R(\boldsymbol{x}) := \boldsymbol{y}^\star$$

  is defined using the **dual** solution

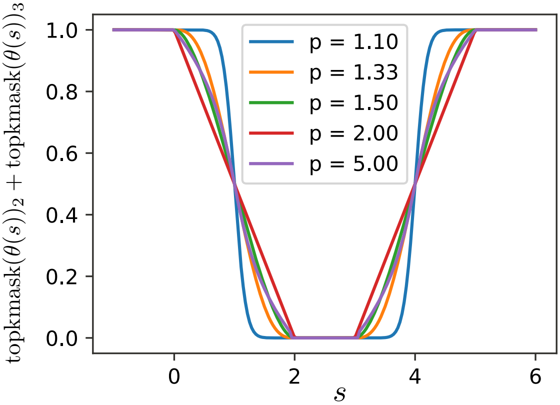
$$\begin{aligned}\boldsymbol{y}^\star &= \underset{\boldsymbol{y}\in\mathcal{C}}{\mathbf{argmax}}\langle \boldsymbol{x}, \boldsymbol{y}\rangle - R(\boldsymbol{y}) \\ &= \underset{\boldsymbol{y}\in\mathbb{R}^n}{\mathbf{argmax}}\langle \boldsymbol{x}, \boldsymbol{y}\rangle - f^*(\boldsymbol{y}) - R(\boldsymbol{y})\end{aligned}$$

- Equivalently, if we define the **primal** solution (infimal convolution)

$$\boldsymbol{u}^\star = \underset{\boldsymbol{u}\in\mathbb{R}^n}{\mathbf{argmin}}\, R^*(\boldsymbol{x} - \boldsymbol{u}) + f(\boldsymbol{u})$$

  then $\boldsymbol{y}^\star = \nabla R^*(\boldsymbol{x} - \boldsymbol{u}^\star)$

Regularized version



$$R(\boldsymbol{y}) = \frac{1}{p}\|\boldsymbol{y}\|_p^p = \frac{1}{p}\sum_{i=1}^{n}|y_i|^p$$

Computing the regularized version

- Recall that the primal solution is

$$\boldsymbol{u}^\star = \operatorname*{\mathbf{argmin}}_{\boldsymbol{u}\in\mathbb{R}^n} R^*(\boldsymbol{x}-\boldsymbol{u}) + f(\boldsymbol{u})$$
$$= \operatorname*{\mathbf{argmin}}_{\boldsymbol{u}\in\mathbb{R}^n} R^*(\boldsymbol{x}-\boldsymbol{u}) + \langle \boldsymbol{u}_{\pi(\boldsymbol{u})}, \mathbf{1}_k \rangle$$

  where $\pi(\boldsymbol{u}) = \mathbf{argsort}(\boldsymbol{u})$
- Reduction to isotonic optimization

$$\boldsymbol{u}^\star_\sigma = \operatorname*{\mathbf{argmin}}_{v_1\geq\cdots\geq v_n} R^*(\boldsymbol{x}_\sigma-\boldsymbol{v}) + f(\boldsymbol{v})$$
$$= \operatorname*{\mathbf{argmin}}_{v_1\geq\cdots\geq v_n} R^*(\boldsymbol{x}_\sigma-\boldsymbol{v}) + \langle \boldsymbol{v}, \mathbf{1}_k \rangle$$

  where $\sigma = \mathbf{argsort}(\boldsymbol{x})$
- Differentiation available in closed form (implicit diff not needed) given $v^\star$

Pool Adjacent Violators (PAV)

$$\operatorname*{\mathbf{argmin}}_{v_1 \geq \cdots \geq v_n} \sum_{i=1}^{n} h_i(v_i)$$

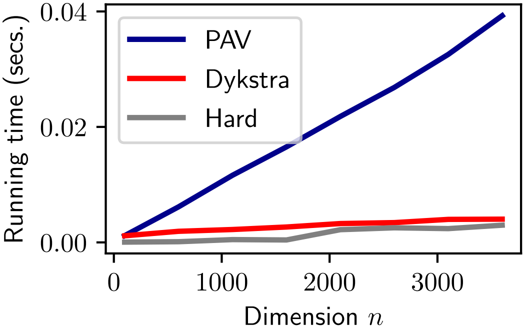
- Partitions the set $[n]$ into disjoint sets $(B_1, \cdots B_m)$, starting from $m = n$ and $B_i = \{i\}$
- Merges these sets until the isotonic condition is met
- Needs to be able to solve $\mathbf{argmin}_{\gamma \in \mathbb{R}} \sum_{i \in B_j} h_i(\gamma)$ in constant time to get $O(n)$ total complexity
  - p-norm regularization case: we need to find the root of a polynomial (easy when $p = 2$ or $p = 4/3$)

Using Dykstra's algorithm

- Key idea: alternate projections between C_1 and C_2

$$\{\boldsymbol{v} \in \mathbb{R}^n : v_1 \geq \cdots \geq v_n\} = \underbrace{\{\boldsymbol{v} \in \mathbb{R}^n : v_1 \geq v_2, v_3 \geq v_4, \ldots\}}_{C_1} \cap \underbrace{\{\boldsymbol{v} \in \mathbb{R}^n : v_2 \geq v_3, v_4 \geq v_5, \ldots\}}_{C_2}$$

- Huge speedup on TPU



Google DeepMind

2 Top-k in magnitude

Top-k in magnitude operator

Same as top-k operator but selects elements with largest **absolute value**

$$\mathbf{topkmag}(\boldsymbol{x}) \coloneqq \boldsymbol{x} \cdot \mathbf{topkmask}(|\boldsymbol{x}|)$$

$$\begin{aligned}\boldsymbol{x} &= (1.7, 3.2, -2.4)\\ \mathbf{top1mag}(\boldsymbol{x}) &= (0.0, 3.2, 0)\\ \mathbf{top2mag}(\boldsymbol{x}) &= (0, 3.2, -2.4)\\ \mathbf{top3mag}(\boldsymbol{x}) &= (1.7, 3.2, -2.4)\end{aligned}$$

Top-k in magnitude as a gradient

- We introduce a **nonlinearity** $\varphi(\boldsymbol{x}) = (\phi(x_1), \ldots, \phi(x_n))$

$$f_\varphi(\boldsymbol{x}) := f(\varphi(\boldsymbol{x})) = \max_{\boldsymbol{y} \in \mathcal{C}} \langle \varphi(\boldsymbol{x}), \boldsymbol{y} \rangle$$

- With $\phi(x) = \frac{1}{2}x^2$, we have

$$\nabla f_\varphi(\boldsymbol{x}) = \textbf{topkmag}(\boldsymbol{x})$$

- With $\phi(x) = x$, we have

$$\nabla f_\varphi(\boldsymbol{x}) = \nabla f(\boldsymbol{x}) = \textbf{topkmask}(\boldsymbol{x})$$

Regularized version

$$\mathbf{topkmag}_R(\boldsymbol{x}) := \boldsymbol{y}^\star = \nabla R^*(\boldsymbol{x} - \boldsymbol{u}^\star)$$

where we defined the **dual** solution

$$\boldsymbol{y}^\star := \underset{\boldsymbol{y} \in \mathbb{R}^n}{\mathbf{argmax}} \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f_\varphi^*(\boldsymbol{y}) - R(\boldsymbol{y})$$

and the **primal** solution (inf-convolution)

$$\boldsymbol{u}^\star = \underset{\boldsymbol{u} \in \mathbb{R}^n}{\mathbf{argmin}}\, R^*(\boldsymbol{x} - \boldsymbol{u}) + f_\varphi(\boldsymbol{u})$$

Conjugate

- For $\varphi(x) = x$: **indicator function** of $\mathcal{C}$

$$f_\varphi^*(\boldsymbol{y}) = f^*(\boldsymbol{y}) = \delta_{\mathcal{C}}(\boldsymbol{y})$$

- For $\varphi(x) = \frac{1}{2}x^2$: **squared k-support norm**

$$f_\varphi^*(\boldsymbol{y}) = \frac{1}{2} \min_{\boldsymbol{z} \in \mathcal{C}} \sum_{i=1}^{n} \frac{y_i^2}{z_i}$$

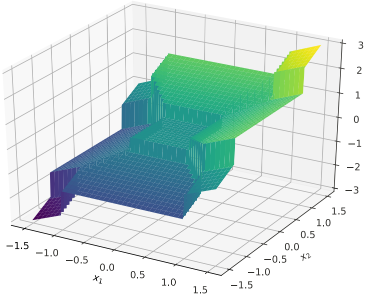
- For general φ: **minimum distance to** $\mathcal{C}$

$$f_\varphi^*(\boldsymbol{y}) = \min_{\boldsymbol{z} \in \mathcal{C}} D_{\phi^*}(\boldsymbol{y}, \boldsymbol{z})$$

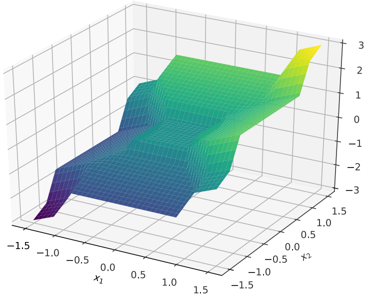
where we defined the **f-divergence**

$$D_f(\boldsymbol{y}, \boldsymbol{z}) := \sum_{i=1}^{n} z_i f(y_i / z_i)$$

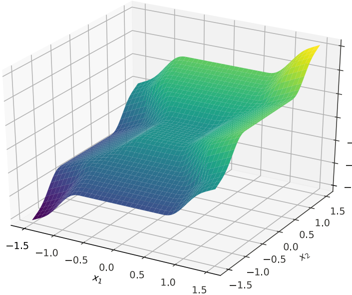
Regularized version



Hard



$p = 2$



$p = 4/3$

Computing the regularized version

- Primal solution

$$\boldsymbol{u}^\star = \mathop{\mathbf{argmin}}_{\boldsymbol{u} \in \mathbb{R}^n} R^*(\boldsymbol{x} - \boldsymbol{u}) + f_\varphi(\boldsymbol{u})$$

- Reduction to isotonic optimization

$$\begin{aligned} \boldsymbol{u}_\sigma^\star &= \mathop{\mathbf{argmin}}_{v_1 \geq \cdots \geq v_n \geq 0} R^*(\boldsymbol{x}_\sigma - \boldsymbol{v}) + f_\varphi(\boldsymbol{v}) \\ &= \mathop{\mathbf{argmin}}_{v_1 \geq \cdots \geq v_n \geq 0} R^*(\boldsymbol{x}_\sigma - \boldsymbol{v}) + f(\varphi(\boldsymbol{v})) \\ &= \mathop{\mathbf{argmin}}_{v_1 \geq \cdots \geq v_n \geq 0} R^*(\boldsymbol{x}_\sigma - \boldsymbol{v}) + \langle \varphi(\boldsymbol{v}), \mathbf{1}_k \rangle \end{aligned}$$

where $\sigma = \mathbf{argsort}(|\boldsymbol{x}|)$ and assuming $\varphi(\boldsymbol{x}) = \varphi(-\boldsymbol{x})$

Nonconvex viewpoint: connection with the ℓ_0 pseudo-norm

- We have

$$f_\varphi(\boldsymbol{x}) = \max_{\boldsymbol{y}\in S_k} \langle \boldsymbol{x}, \boldsymbol{y}\rangle - \sum_{i=1}^n \phi(y_i)$$

where

$$\varphi(x) = (\phi(x_1), \dots, \phi(x_n)) \quad \text{and} \quad S_k := \{\boldsymbol{y} \in \mathbb{R}^n \colon \|\boldsymbol{y}\|_0 \le k\}$$

- $f_\varphi^*(\boldsymbol{y})$ is the **convex envelope** of

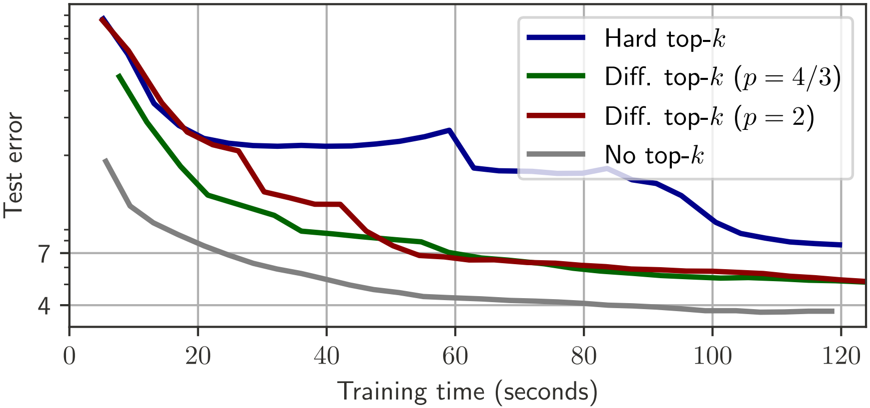
$$\boldsymbol{y} \mapsto \sum_{i=1}^n \phi^*(y_i) + \delta_{S_k}(\boldsymbol{y})$$

With $\phi(x) = \frac{1}{2}x^2$, $f_\varphi^*(\boldsymbol{y})$ is the **squared k-support norm**

Google DeepMind

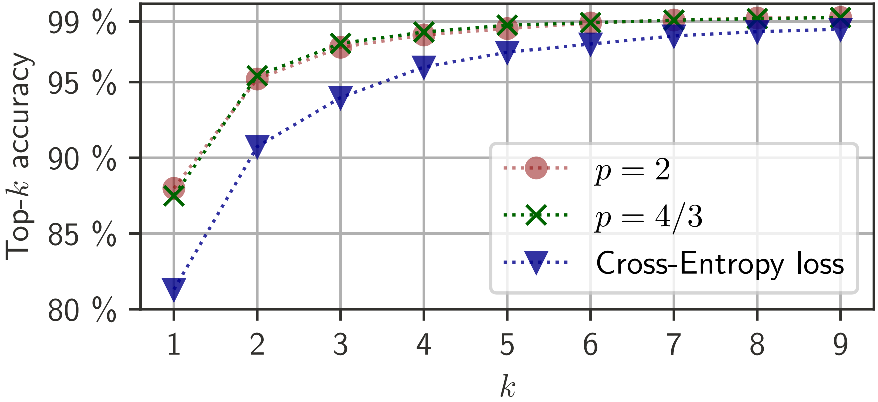
3 Applications

Weight pruning (multilayer perceptron, MNIST)



$$W_i \leftarrow \textbf{topkmag}(W_i), i \in \{2, 3\}$$
$$W_3\sigma(W_2\sigma(W_1\boldsymbol{a} + \boldsymbol{b}_1) + \boldsymbol{b}_2) + \boldsymbol{b}_3$$

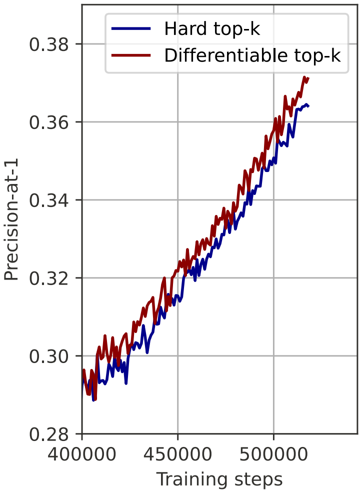
Top-k classification (vision transformer, CIFAR 100)



$$\ell_R(\boldsymbol{\theta}, \boldsymbol{t}) := [\mathbf{max}_{\boldsymbol{y} \in \mathcal{C}} \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle - R(\boldsymbol{y})] - \langle \boldsymbol{\theta}, \boldsymbol{t} \rangle$$

$\boldsymbol{\theta}$: logits, $\boldsymbol{t}$: target

Sparse Mixture of Vision Transformers

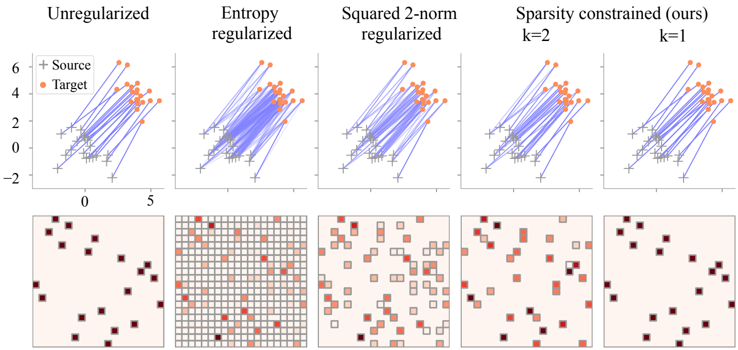


JFT-300M dataset (305 million images)

Sparsity-constrained OT

Sparsity-constrained OT.
Liu, Puigcerver, Blondel.
ICLR, 2023.

$$\min_{T \in \mathcal{U}(a,b)} \langle T, C \rangle + \sum_{j=1}^{n} f_{\varphi}^{*}(\boldsymbol{t}_j)$$



Google DeepMind

Thank you!