

# 3. Computing OT for data sciences

- Reality check
- Regularizations
- Entropic regularization

# What matters for practitioners?

i.i.d samples  $\mathbf{x}_1, \dots, \mathbf{x}_n \sim \boldsymbol{\mu}$ ,  $\mathbf{y}_1, \dots, \mathbf{y}_m \sim \boldsymbol{\nu}$ ,

$$\hat{\boldsymbol{\mu}}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{\mathbf{x}_i}, \hat{\boldsymbol{\nu}}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{\mathbf{y}_j}$$

## Computational properties

Compute/approximate  $W_p(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\nu}}_m)$ ?

## Statistical properties

$\mathbb{E} [|W_p(\boldsymbol{\mu}, \boldsymbol{\nu}) - W_p(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\nu}}_m)|] \leq f(n, m)$ ?

# What matters for practitioners?

i.i.d samples  $\mathbf{x}_1, \dots, \mathbf{x}_n \sim \boldsymbol{\mu}$ ,  $\mathbf{y}_1, \dots, \mathbf{y}_m \sim \boldsymbol{\nu}$ ,

$$\hat{\boldsymbol{\mu}}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{\mathbf{x}_i}, \hat{\boldsymbol{\nu}}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{\mathbf{y}_j}$$

Computational properties



$$O((n + m)nm \log(n + m))$$

Statistical properties

$$\mathbb{E} [|W_p(\boldsymbol{\mu}, \boldsymbol{\nu}) - W_p(\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\nu}}_m)|] \leq f(n, m)?$$

# Sample Complexity



If  $\Omega = \mathbb{R}^d, d > 3$

$$\mathbb{E} [|W_p(\mu, \nu) - W_p(\hat{\mu}_n, \hat{\nu}_n)|] = O(n^{-1/d})$$

- **[Dudley'69][Dereich+'11][Fournier+'13] & others..**
- **[Weed+'17]**: sharper results when measures' support has “low effective  $d$ ” in metric spaces
- **[Weed+'19]** for smooth densities
- **Lower bounds**: optimal quantization error.

# From theory to practice ?

$$\hat{\mu}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{x_i}, \hat{\nu}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{y_j}$$

Computational properties



$$O((n + m)nm \log(n + m))$$

Statistical properties



$$\mathbb{E} [|W_p(\mu, \nu) - W_p(\hat{\mu}_n, \hat{\nu}_m)|] = O(n^{-1/d})$$

# From theory to practice ?

$$\hat{\mu}_n \stackrel{\text{def}}{=} \frac{1}{n} \sum_i \delta_{x_i}, \hat{\nu}_m \stackrel{\text{def}}{=} \frac{1}{m} \sum_j \delta_{y_j}$$

Computational properties



$$O((n + m)nm \log(n + m))$$

Statistical properties



$$\mathbb{E} [|W_p(\mu, \nu) - W_p(\hat{\mu}_n, \hat{\nu}_m)|] = O(n^{-1/d})$$

If we use OT in data sciences we must regularize the problem to improve on both aspects!

# Many ways to regularize (dual) OT

$$\sup_{\varphi(x) + \psi(y) \leq c(x, y)} \int \varphi d\mu + \int \psi d\nu.$$

# Many ways to regularize (dual) OT

$$\sup_{\varphi(x) + \psi(y) \leq c(x, y)} \int \varphi d\mu + \int \psi d\nu.$$

- RKHS for potentials [GCBP'16], dualize/smooth indicator constraint

# Many ways to regularize (dual) OT

$$\sup_{\varphi(x) + \psi(y) \leq c(x, y)} \int \varphi d\mu + \int \psi d\nu.$$

- RKHS for potentials [**GCBP'16**], dualize/smooth indicator constraint

$$W_1(\mu, \nu) = \sup_{\varphi \text{ 1-Lipschitz}} \int \varphi(d\mu - d\nu).$$

- Parameterize functions using ReLU Deep net with bounded weights [**ACB'17**] or use Wavelet decompositions [**Shirdhonkar+'08**] for low  $d$ .

# Many ways to regularize (primal) OT

$$\inf_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \iint \mathbf{c}(x, y) \mathbf{P}(dx, dy).$$

# Many ways to regularize (primal) OT

$$\inf_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \iint \mathbf{c}(x, y) \mathbf{P}(dx, dy).$$

- Change cost function: threshold metric [Pele+’09], use geodesic distance on graphs [Beckman’52] [Lin+’07], [Solomon+’14], simplifies the LP.

# Many ways to regularize (primal) OT

$$\inf_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \iint \mathbf{c}(x, y) \mathbf{P}(dx, dy).$$

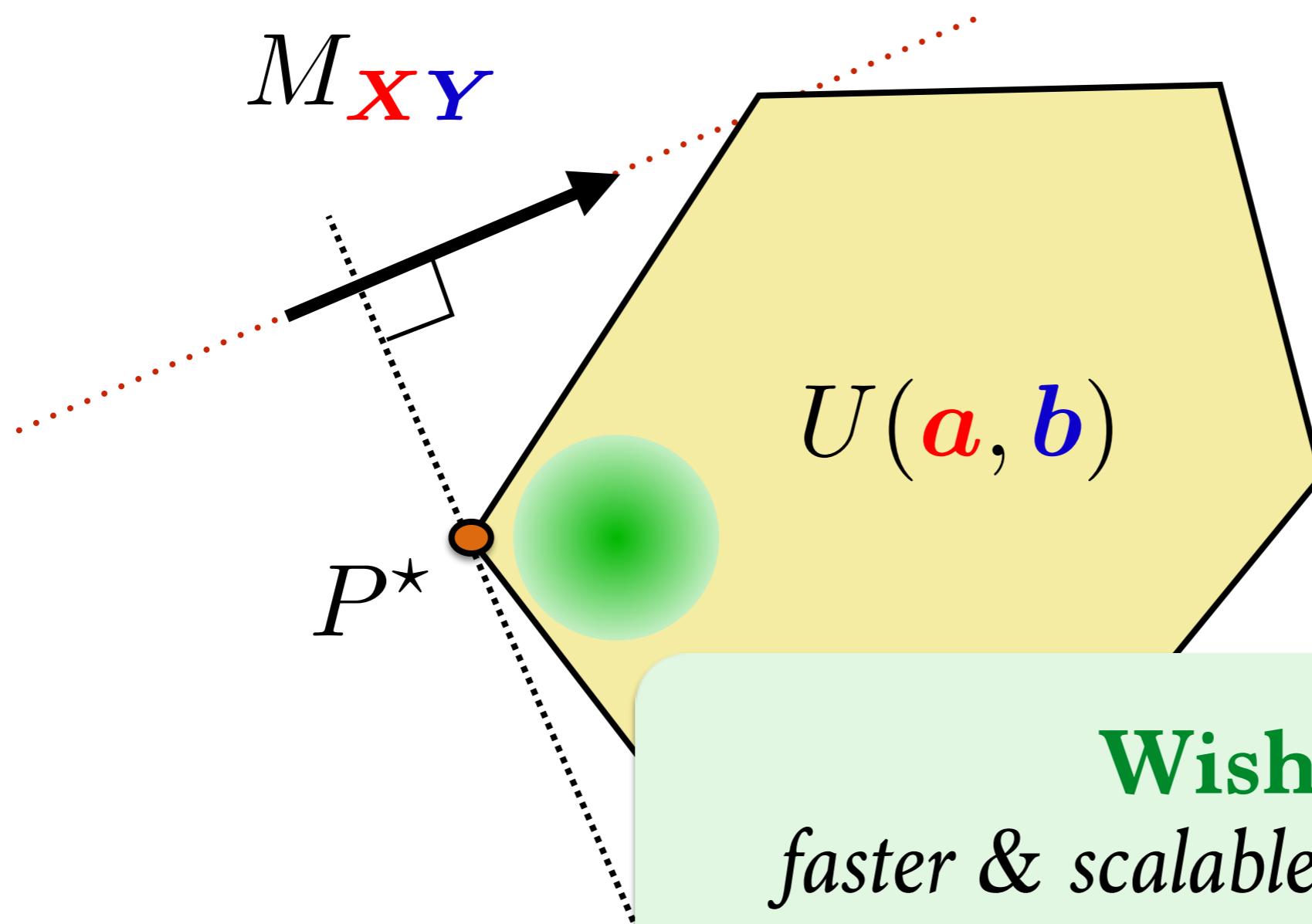
- Change cost function: threshold metric [Pele+’09], use geodesic distance on graphs [Beckman’52] [Lin+’07], [Solomon+’14], simplifies the LP.
- Change measures: Quantize first [Canas+’12]; use Gaussians [Gelbrich’92]; projections on random lines [Rabin+’11] &  $k$ -dimensional subspaces [PC’19][Weed+’19].

# Many ways to regularize (primal) OT

$$\inf_{\mathbf{P} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \iint c(x, y) \mathbf{P}(dx, dy).$$

- Change cost function: threshold metric [Pele+’09], use geodesic distance on graphs [Beckman’52] [Lin+’07], [Solomon+’14], simplifies the LP.
- Change measures: Quantize first [Canas+’12]; use Gaussians [Gelbrich’92]; projections on random lines [Rabin+’11] &  $k$ -dimensional subspaces [PC’19][Weed+’19].
- Add prior on coupling, e.g. (entropic) regularization [C’13][GP’16][GCBCP’19][JMPC’20]

# Regularization on the Primal



## Wishlist:

*faster & scalable, more stable,  
(automatically) differentiable*

# Entropic Regularization [Wilson'69]

Def. Regularized Wasserstein,  $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle - \gamma E(\boldsymbol{P})$$

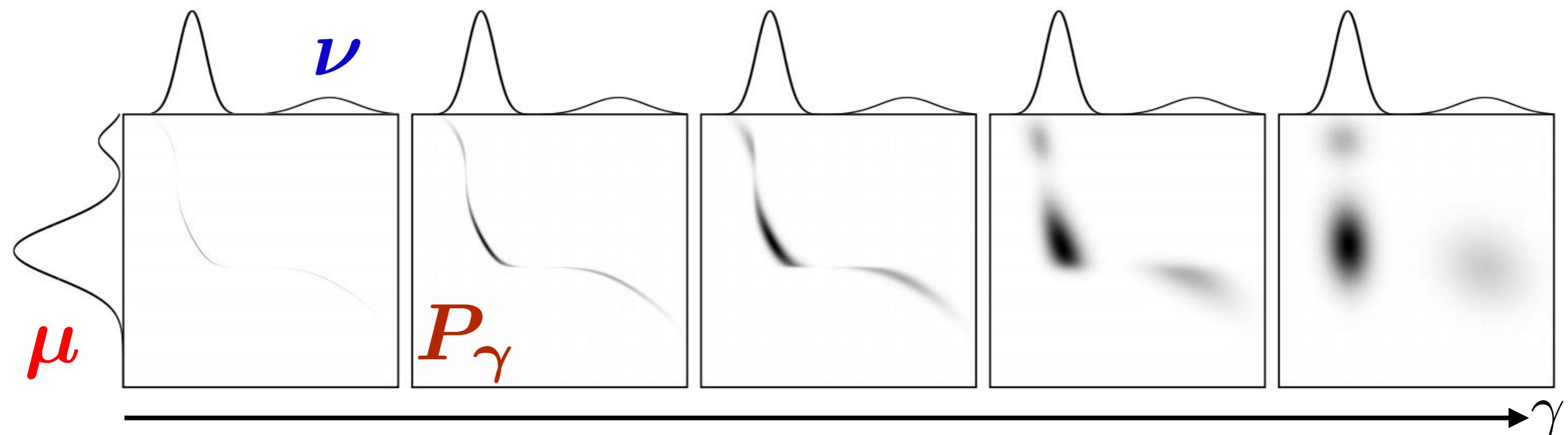
$$E(P) \stackrel{\text{def}}{=} - \sum_{i,j=1}^{nm} P_{ij} (\log P_{ij} - 1)$$

**Note: Unique** optimal solution because of strong concavity of entropy

# Entropic Regularization [Wilson'69]

Def. Regularized Wasserstein,  $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle - \gamma E(\boldsymbol{P})$$

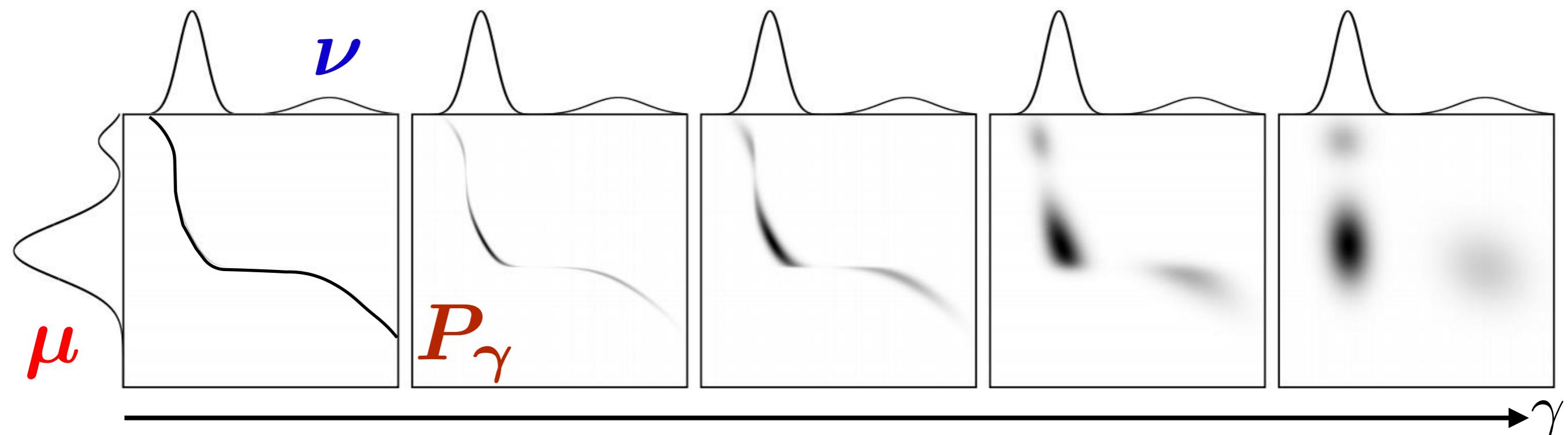


Note: Unique optimal solution because of strong concavity of entropy

# Entropic Regularization [Wilson'69]

Def. Regularized Wasserstein,  $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle - \gamma E(\boldsymbol{P})$$

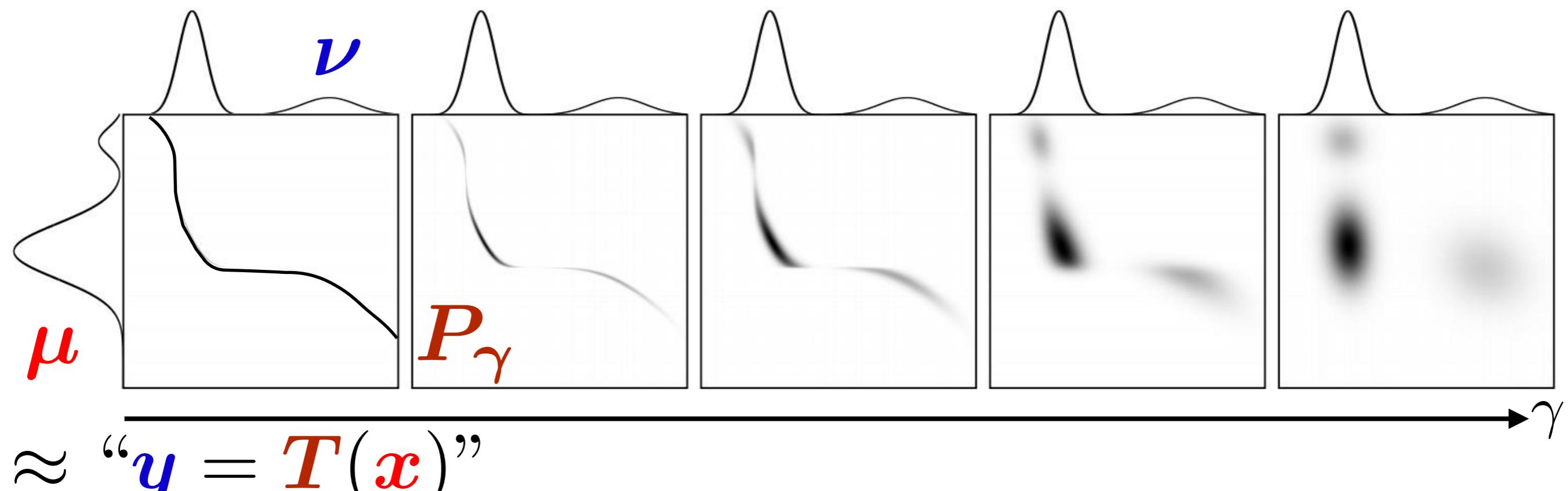


Note: Unique optimal solution because of strong concavity of entropy

# Entropic Regularization [Wilson'69]

Def. Regularized Wasserstein,  $\gamma \geq 0$

$$W_\gamma(\boldsymbol{\mu}, \boldsymbol{\nu}) \stackrel{\text{def}}{=} \min_{\boldsymbol{P} \in U(\boldsymbol{a}, \boldsymbol{b})} \langle \boldsymbol{P}, M_{\boldsymbol{X}\boldsymbol{Y}} \rangle - \gamma E(\boldsymbol{P})$$



$\approx “\boldsymbol{y} = \mathbf{T}(\boldsymbol{x})”$

Note: Unique optimal solution because of strong concavity of entropy

# Fast & Scalable Algorithm

**Prop.** If  $P_\gamma \stackrel{\text{def}}{=} \underset{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})}{\operatorname{argmin}} \langle \mathbf{P}, M_{\mathbf{X}\mathbf{Y}} \rangle - \gamma E(\mathbf{P})$

then  $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$ , such that

$$P_\gamma = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}), \quad \mathbf{K} \stackrel{\text{def}}{=} e^{-M_{\mathbf{X}\mathbf{Y}}} / \gamma$$

# Fast & Scalable Algorithm

**Prop.** If  $P_\gamma \stackrel{\text{def}}{=} \underset{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})}{\operatorname{argmin}} \langle \mathbf{P}, M_{\mathbf{X}\mathbf{Y}} \rangle - \gamma E(\mathbf{P})$

then  $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$ , such that

$$P_\gamma = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}), \quad \mathbf{K} \stackrel{\text{def}}{=} e^{-M_{\mathbf{X}\mathbf{Y}}} / \gamma$$

$$L(P, \alpha, \beta) = \sum_{ij} P_{ij} M_{ij} + \gamma P_{ij} (\log P_{ij} - 1) + \alpha^T (P \mathbf{1} - \mathbf{a}) + \beta^T (P^T \mathbf{1} - \mathbf{b})$$

$$\partial L / \partial P_{ij} = M_{ij} + \gamma \log P_{ij} + \alpha_i + \beta_j$$

$$(\partial L / \partial P_{ij} = 0) \Rightarrow P_{ij} = e^{\frac{\alpha_i}{\gamma}} e^{-\frac{M_{ij}}{\gamma}} e^{\frac{\beta_j}{\gamma}} = \mathbf{u}_i K_{ij} \mathbf{v}_j$$

# Fast & Scalable Algorithm

Prop. If  $P_\gamma \stackrel{\text{def}}{=} \underset{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})}{\operatorname{argmin}} \langle \mathbf{P}, M_{\mathbf{X}\mathbf{Y}} \rangle - \gamma E(\mathbf{P})$

then  $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$ , such that

$$P_\gamma = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}), \quad \mathbf{K} \stackrel{\text{def}}{=} e^{-M_{\mathbf{X}\mathbf{Y}}} / \gamma$$

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}) \mathbf{1}_m = \mathbf{a} \\ \operatorname{diag}(\mathbf{v}) \mathbf{K}^T \operatorname{diag}(\mathbf{u}) \mathbf{1}_n = \mathbf{b} \end{cases}$$

# Fast & Scalable Algorithm

**Prop.** If  $P_\gamma \stackrel{\text{def}}{=} \underset{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})}{\operatorname{argmin}} \langle \mathbf{P}, M_{\mathbf{X}\mathbf{Y}} \rangle - \gamma E(\mathbf{P})$

then  $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$ , such that

$$P_\gamma = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}), \quad \mathbf{K} \stackrel{\text{def}}{=} e^{-M_{\mathbf{X}\mathbf{Y}}} / \gamma$$

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}) \mathbf{1}_m = \mathbf{a} \\ \operatorname{diag}(\mathbf{v}) \underbrace{\mathbf{K}^T \operatorname{diag}(\mathbf{u})}_{\mathbf{u}} \mathbf{1}_n = \mathbf{b} \end{cases}$$

# Fast & Scalable Algorithm

**Prop.** If  $P_\gamma \stackrel{\text{def}}{=} \underset{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})}{\operatorname{argmin}} \langle \mathbf{P}, M_{\mathbf{XY}} \rangle - \gamma E(\mathbf{P})$

then  $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$ , such that

$$P_\gamma = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}), \quad \mathbf{K} \stackrel{\text{def}}{=} e^{-M_{\mathbf{XY}}} / \gamma$$

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \operatorname{diag}(\mathbf{u}) \mathbf{K} \underbrace{\operatorname{diag}(\mathbf{v}) \mathbf{1}_m}_{\mathbf{v}} = \mathbf{a} \\ \operatorname{diag}(\mathbf{v}) \mathbf{K}^T \underbrace{\operatorname{diag}(\mathbf{u}) \mathbf{1}_n}_{\mathbf{u}} = \mathbf{b} \end{cases}$$

# Fast & Scalable Algorithm

Prop. If  $P_\gamma \stackrel{\text{def}}{=} \underset{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})}{\operatorname{argmin}} \langle \mathbf{P}, M_{\mathbf{X}\mathbf{Y}} \rangle - \gamma E(\mathbf{P})$

then  $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$ , such that

$$P_\gamma = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}), \quad \mathbf{K} \stackrel{\text{def}}{=} e^{-M_{\mathbf{X}\mathbf{Y}}} / \gamma$$

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \operatorname{diag}(\mathbf{u}) \mathbf{K} \mathbf{v} \\ \operatorname{diag}(\mathbf{v}) \mathbf{K}^T \mathbf{u} \end{cases} = \mathbf{a} \\ = \mathbf{b}$$

# Fast & Scalable Algorithm

**Prop.** If  $P_\gamma \stackrel{\text{def}}{=} \underset{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})}{\operatorname{argmin}} \langle \mathbf{P}, M_{\mathbf{XY}} \rangle - \gamma E(\mathbf{P})$

then  $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$ , such that

$$P_\gamma = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}), \quad \mathbf{K} \stackrel{\text{def}}{=} e^{-M_{\mathbf{XY}}} / \gamma$$

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \mathbf{u} \odot \mathbf{K} \mathbf{v} \\ \mathbf{v} \odot \mathbf{K}^T \mathbf{u} \end{cases} \begin{matrix} = \mathbf{a} \\ = \mathbf{b} \end{matrix}$$

# Fast & Scalable Algorithm

**Prop.** If  $P_\gamma \stackrel{\text{def}}{=} \underset{\mathbf{P} \in U(\mathbf{a}, \mathbf{b})}{\operatorname{argmin}} \langle \mathbf{P}, M_{\mathbf{X}\mathbf{Y}} \rangle - \gamma E(\mathbf{P})$

then  $\exists! \mathbf{u} \in \mathbb{R}_+^n, \mathbf{v} \in \mathbb{R}_+^m$ , such that

$$P_\gamma = \operatorname{diag}(\mathbf{u}) \mathbf{K} \operatorname{diag}(\mathbf{v}), \quad \mathbf{K} \stackrel{\text{def}}{=} e^{-M_{\mathbf{X}\mathbf{Y}}} / \gamma$$

$$P_\gamma \in U(\mathbf{a}, \mathbf{b}) \Leftrightarrow \begin{cases} \mathbf{u} = \mathbf{a} / \mathbf{K} \mathbf{v} \\ \mathbf{v} = \mathbf{b} / \mathbf{K}^T \mathbf{u} \end{cases}$$

# Fast & Scalable Algorithm

Sinkhorn's Algorithm : Repeat

1.  $\mathbf{u} = \mathbf{a}/K\mathbf{v}$
2.  $\mathbf{v} = \mathbf{b}/K^T \mathbf{u}$

# Fast & Scalable Algorithm

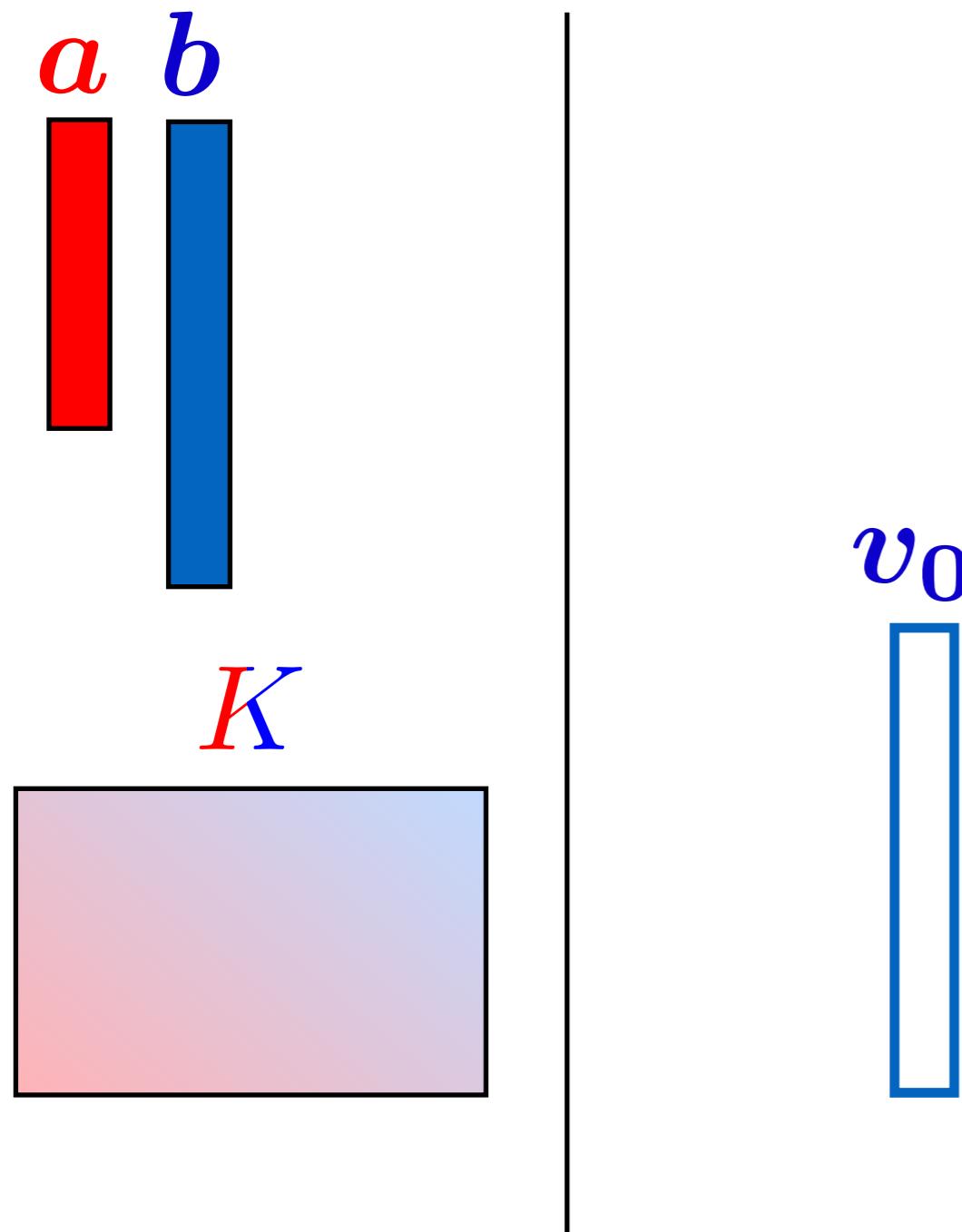
## Sinkhorn's Algorithm : Repeat

1.  $\mathbf{u} = \mathbf{a}/\mathbf{K}\mathbf{v}$
2.  $\mathbf{v} = \mathbf{b}/\mathbf{K}^T \mathbf{u}$

- [Sinkhorn'64] proved first convergence result  
[Franklin+'89] characterised linear convergence
- Recent wave of great results by [Altschuler+'17]  
[Dvurechensky+18][Lin+19]
- $O(nm)$  complexity, GPGPU parallel [C'13].
- $O(n \log n)$  on gridded spaces using convolutions.  
[Solomon'+15]

# Fast & Scalable Algorithm

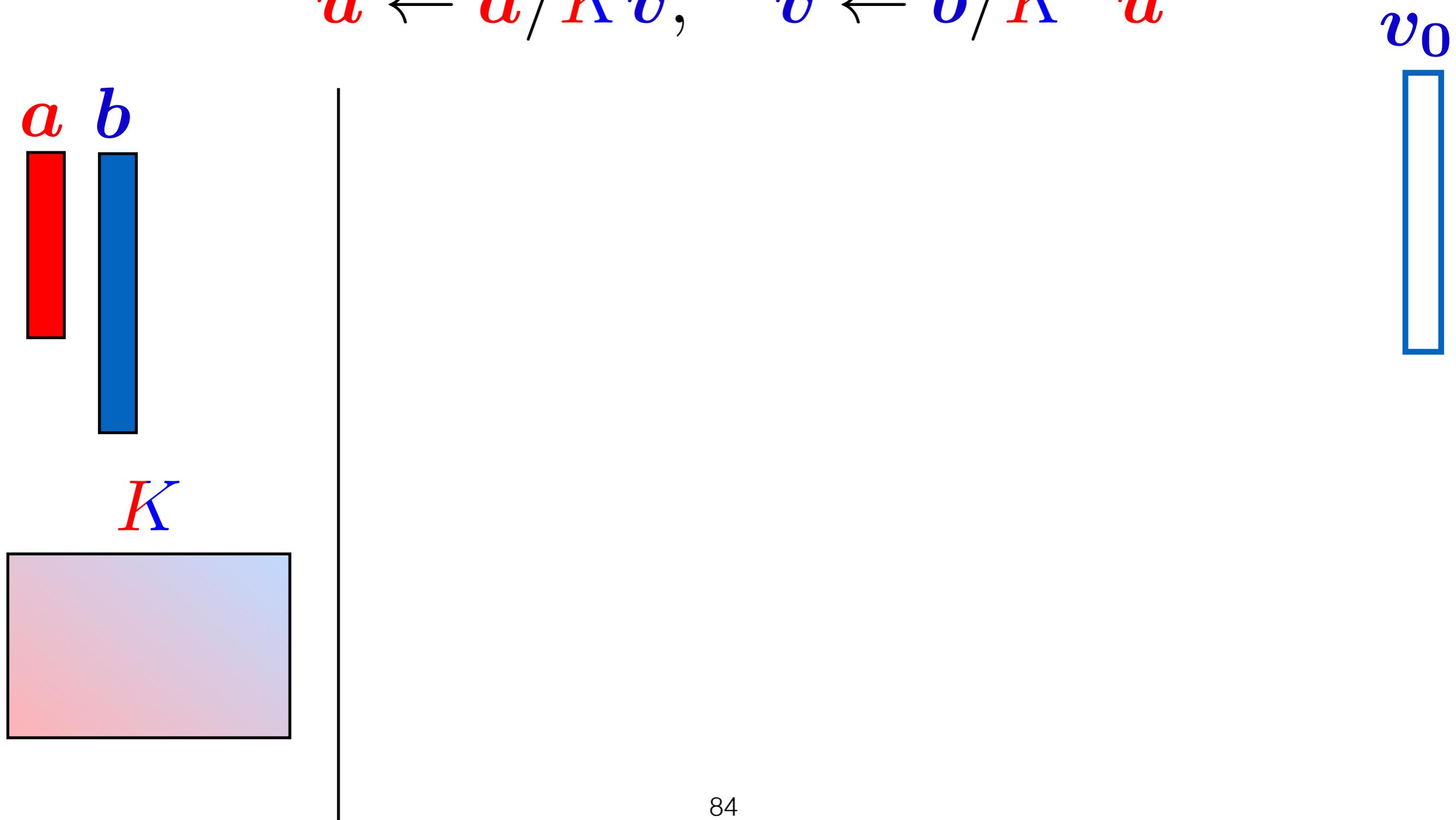
- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$   
 $\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

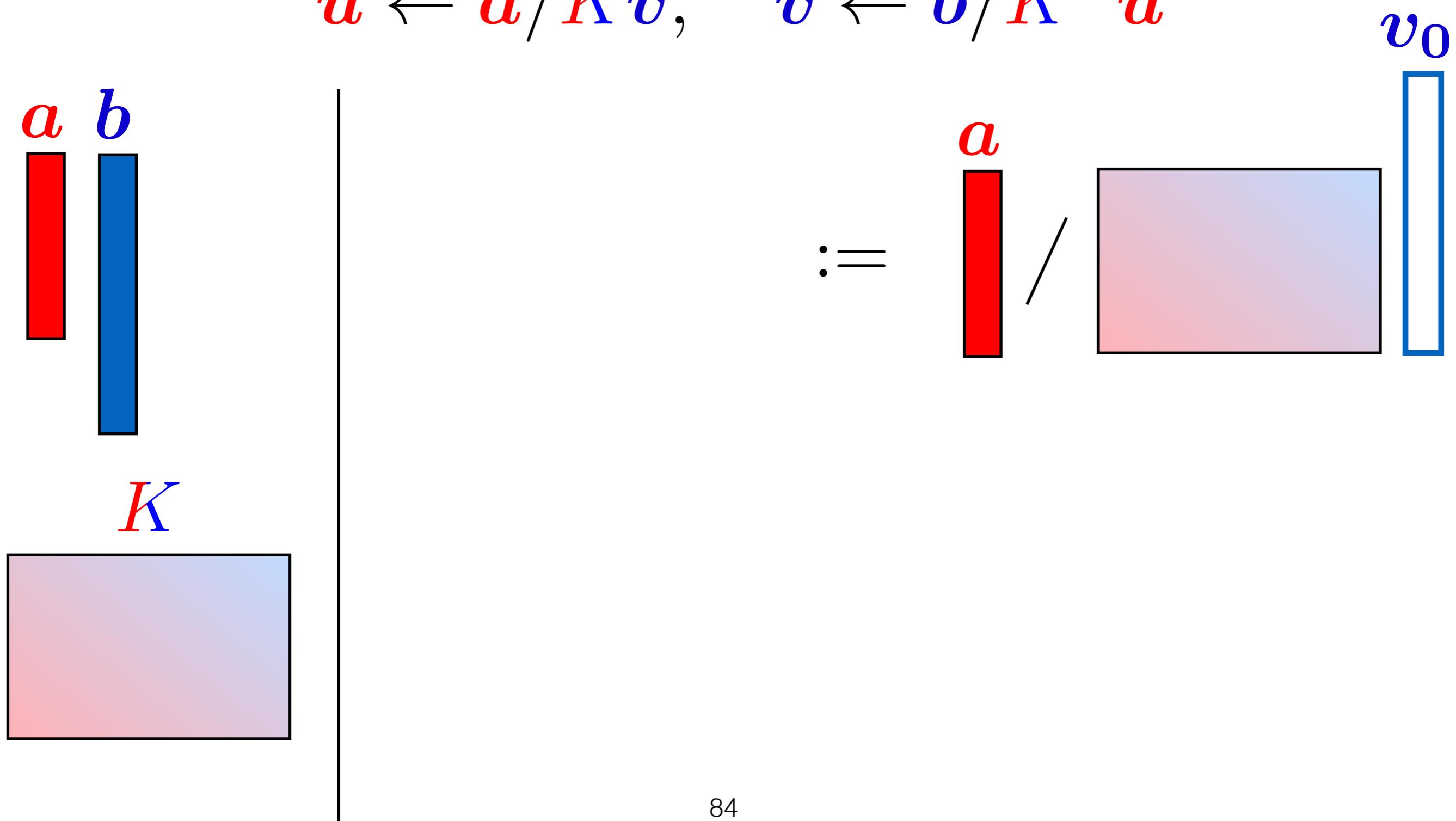
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

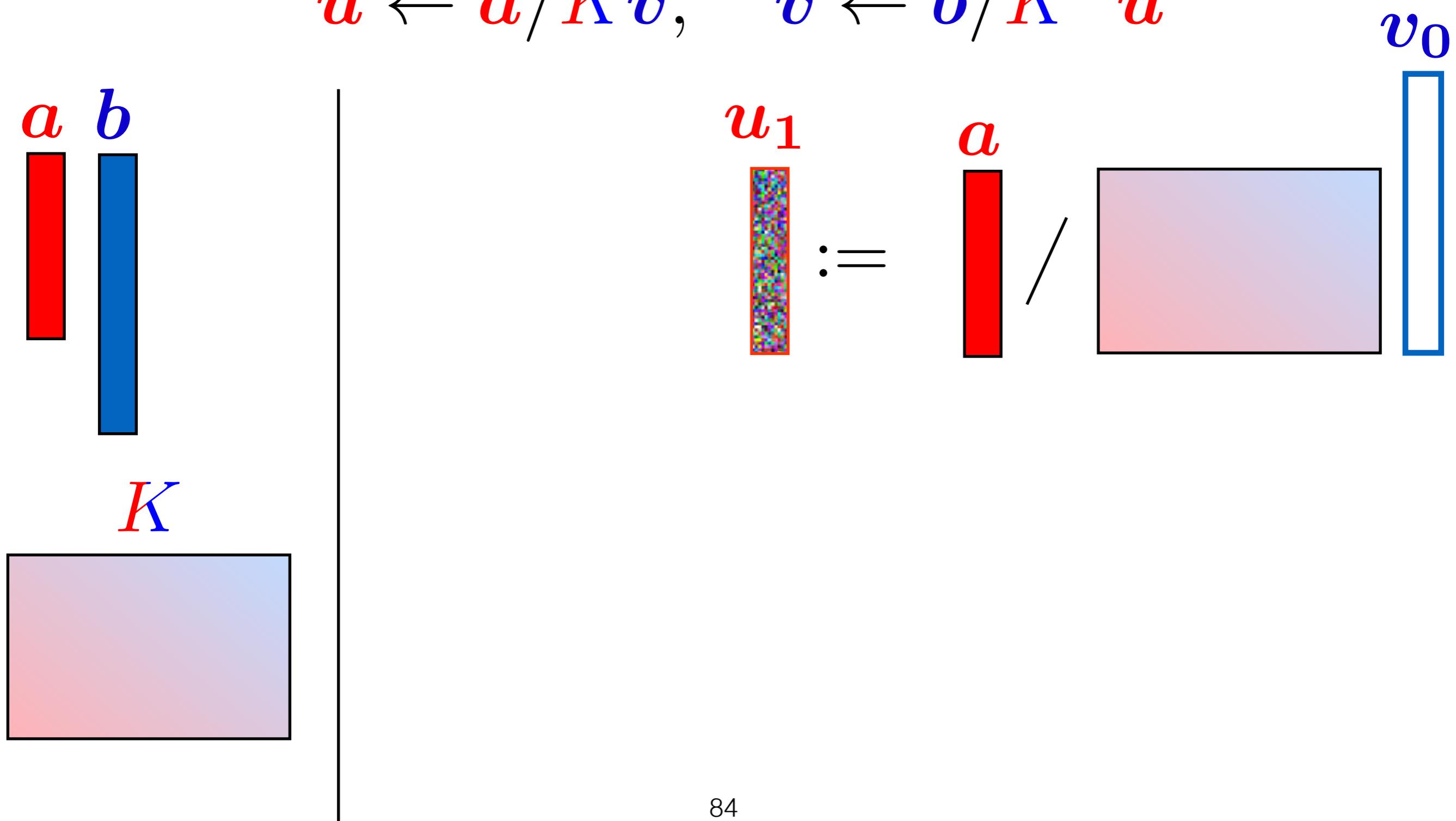
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

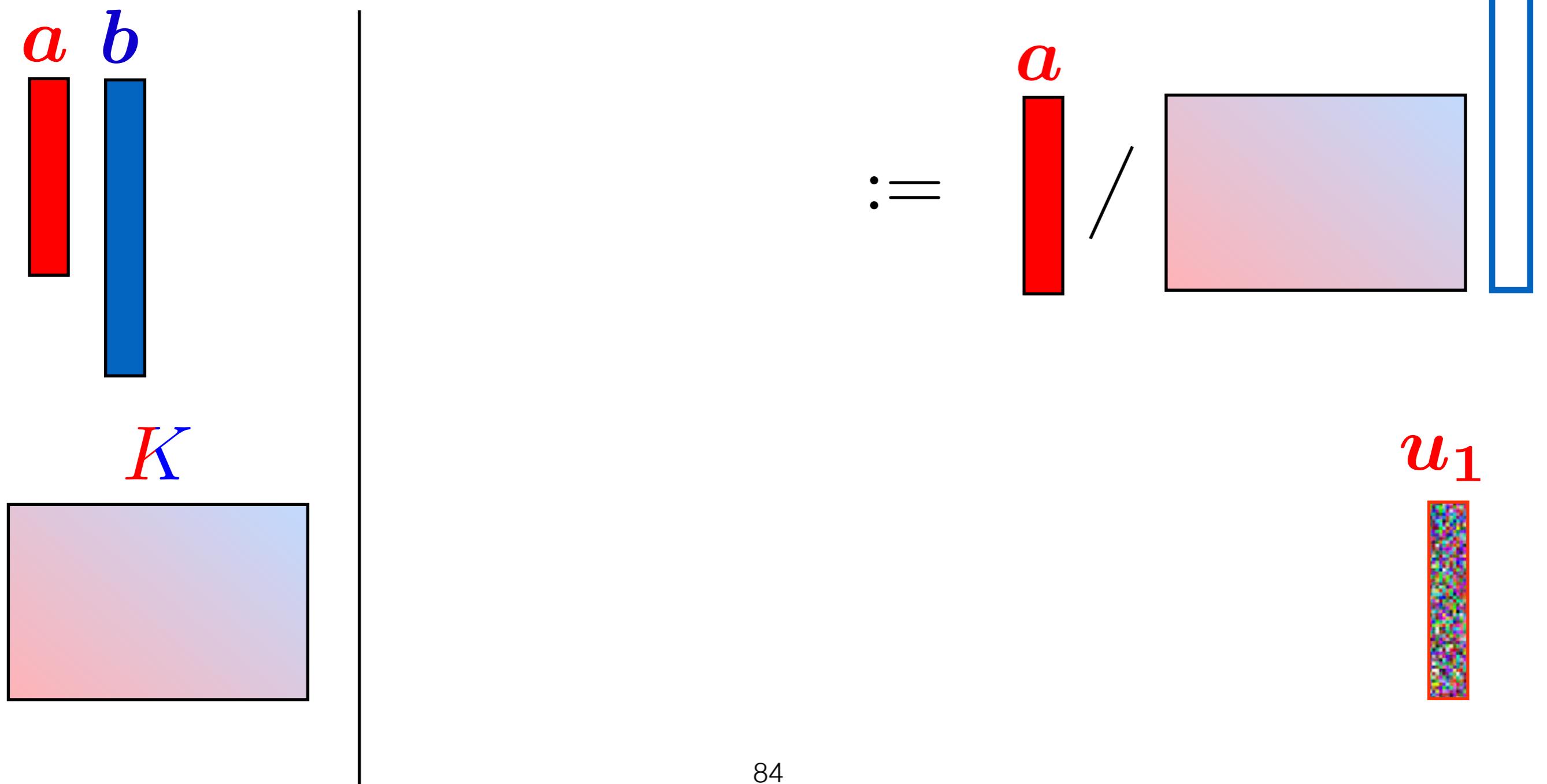
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

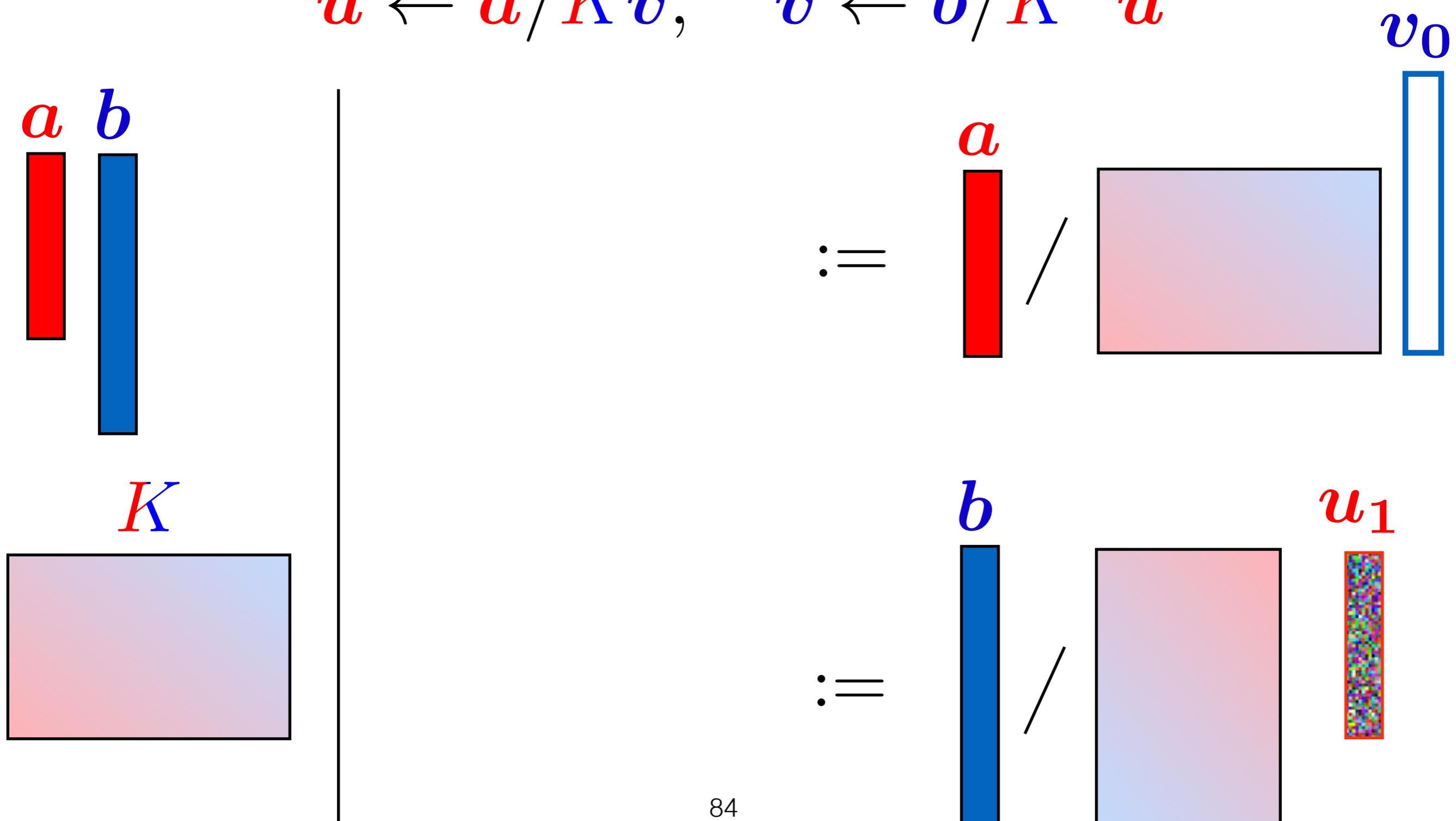
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

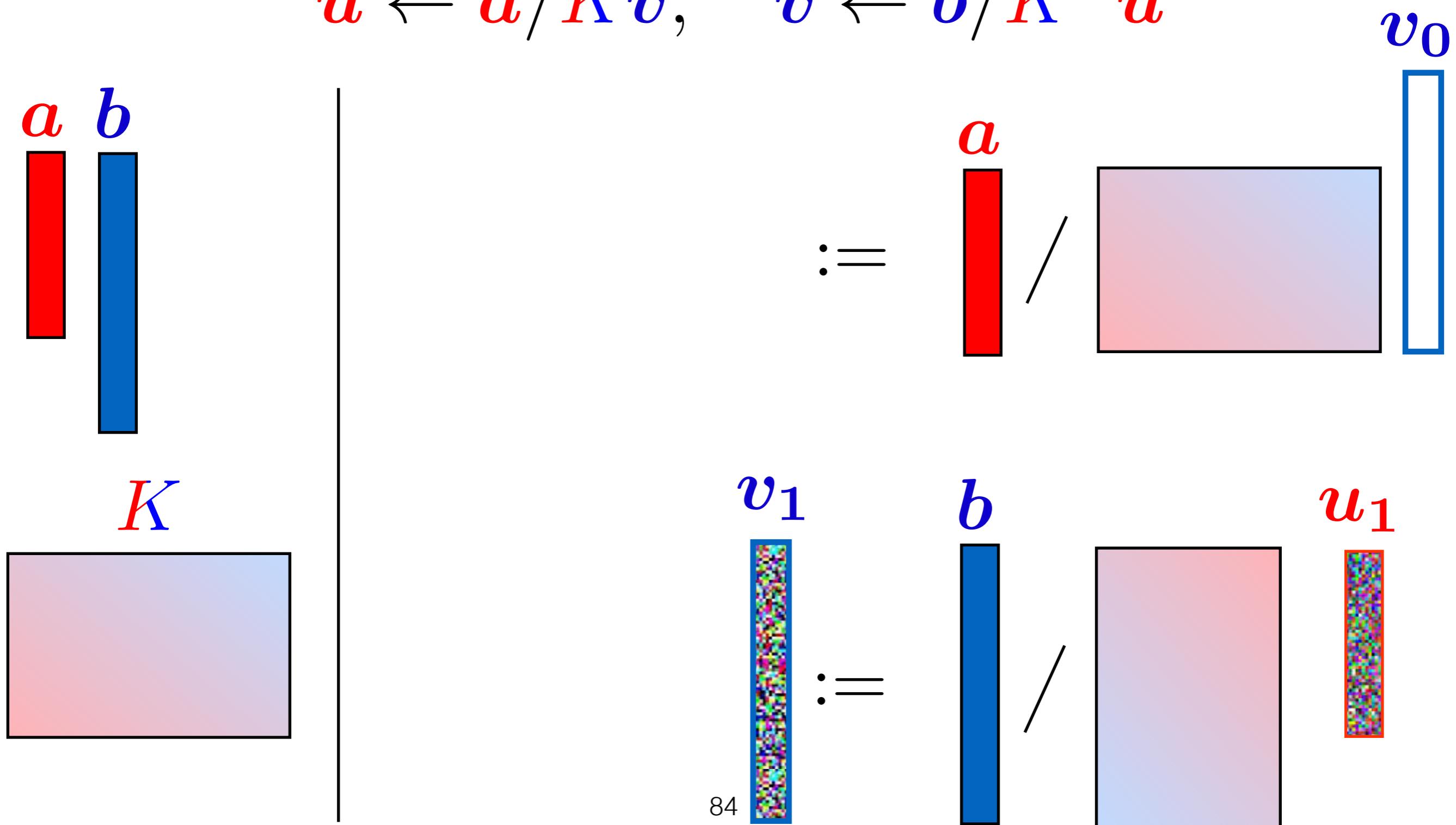
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$

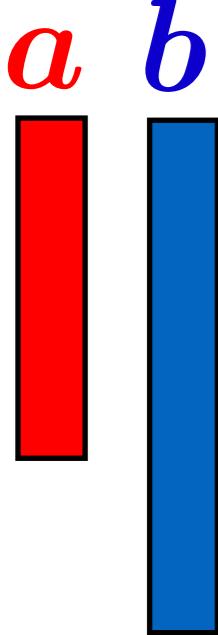


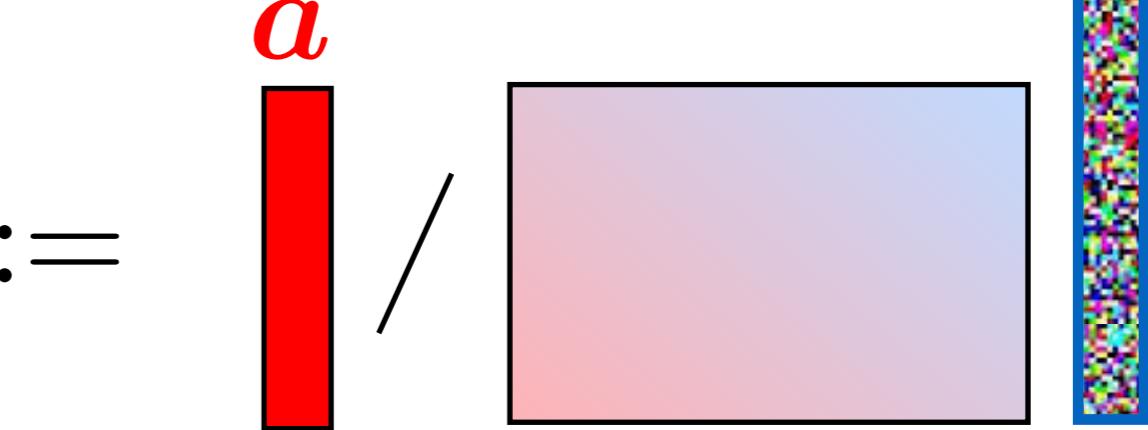
# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

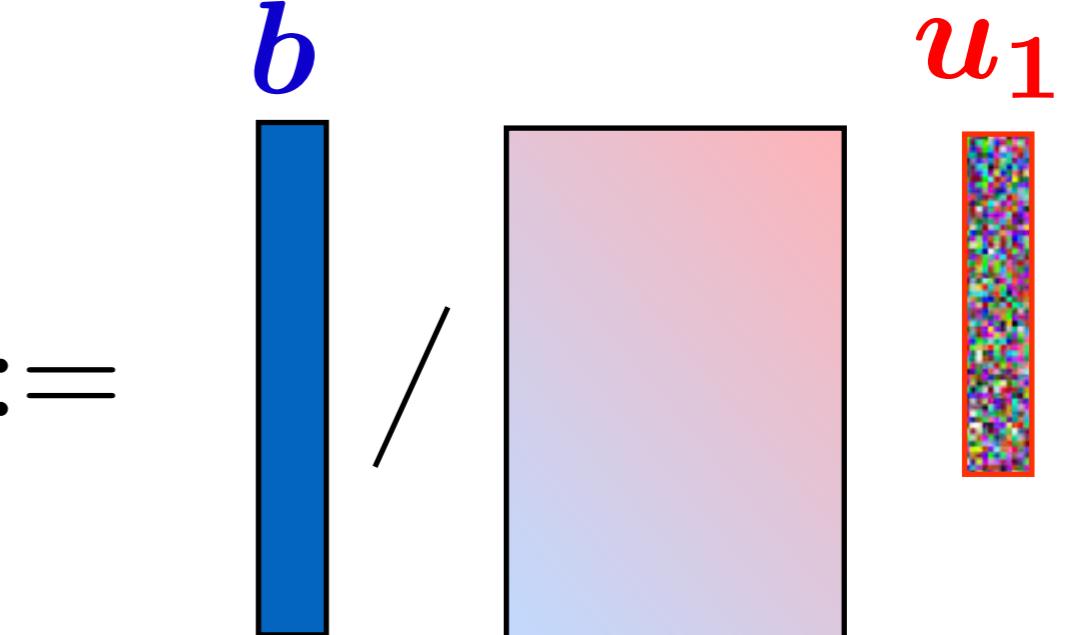
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$

$v_1$

$$\begin{matrix} \mathbf{a} & \mathbf{b} \end{matrix}$$


$$:= \begin{matrix} \mathbf{a} \\ \mathbf{b} \end{matrix} / \begin{matrix} \text{pink gradient} \\ \text{blue gradient} \end{matrix} v_1$$


$$\mathbf{K}$$

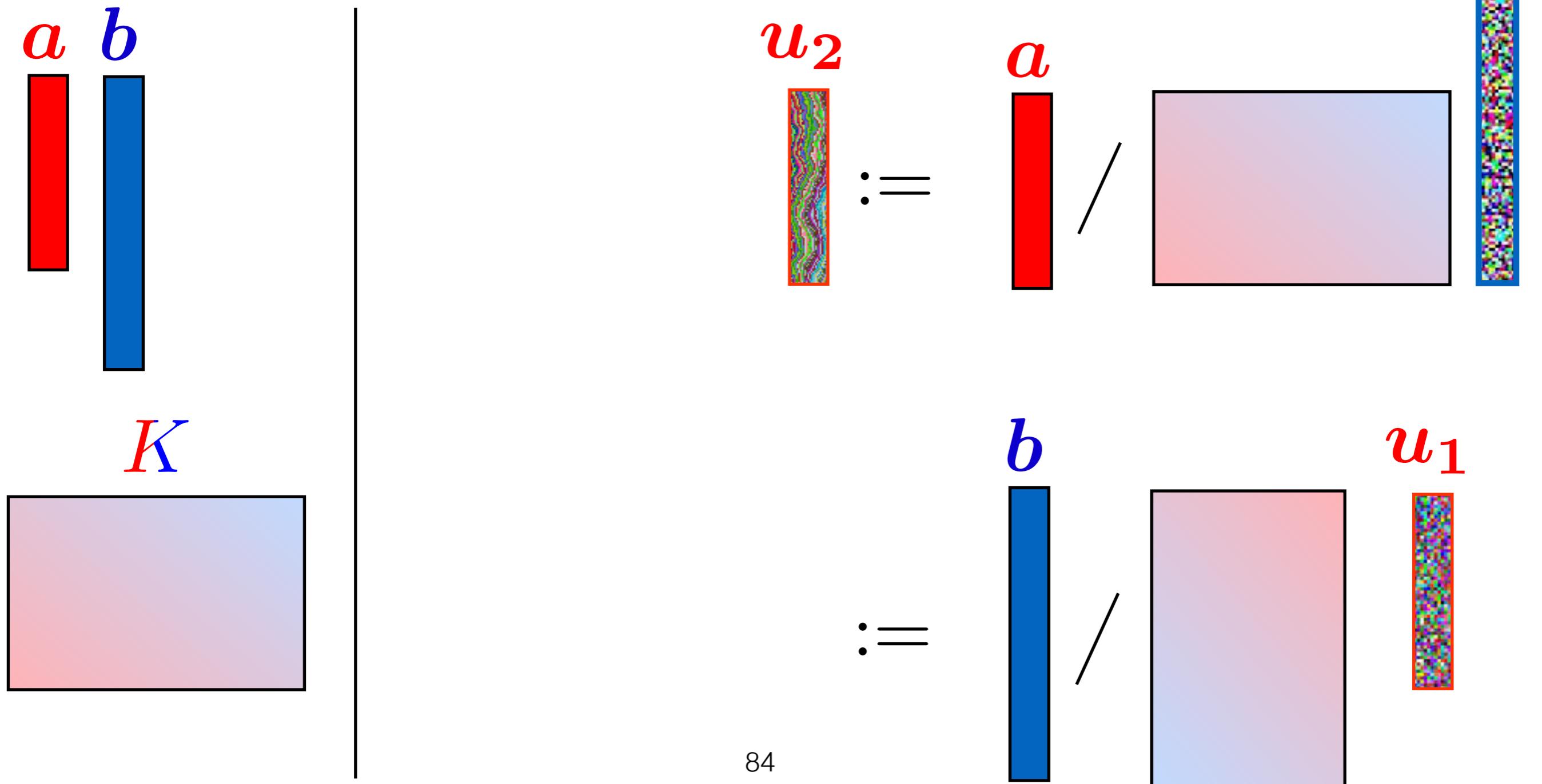

$$:= \begin{matrix} \mathbf{b} \\ \mathbf{a} \end{matrix} / \begin{matrix} \text{blue gradient} \\ \text{pink gradient} \end{matrix} u_1$$


# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$

$v_1$

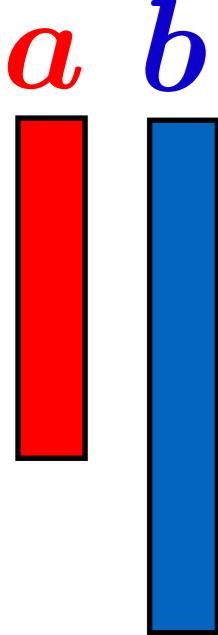


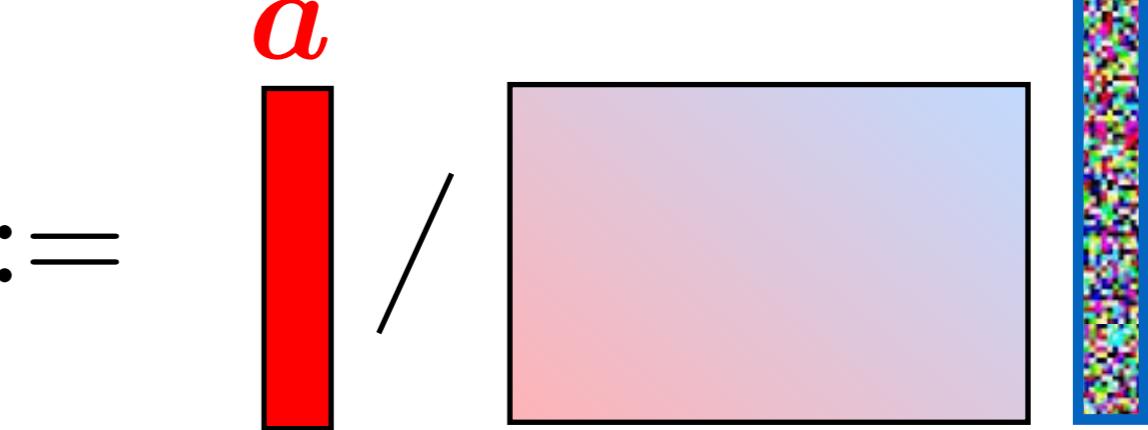
# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

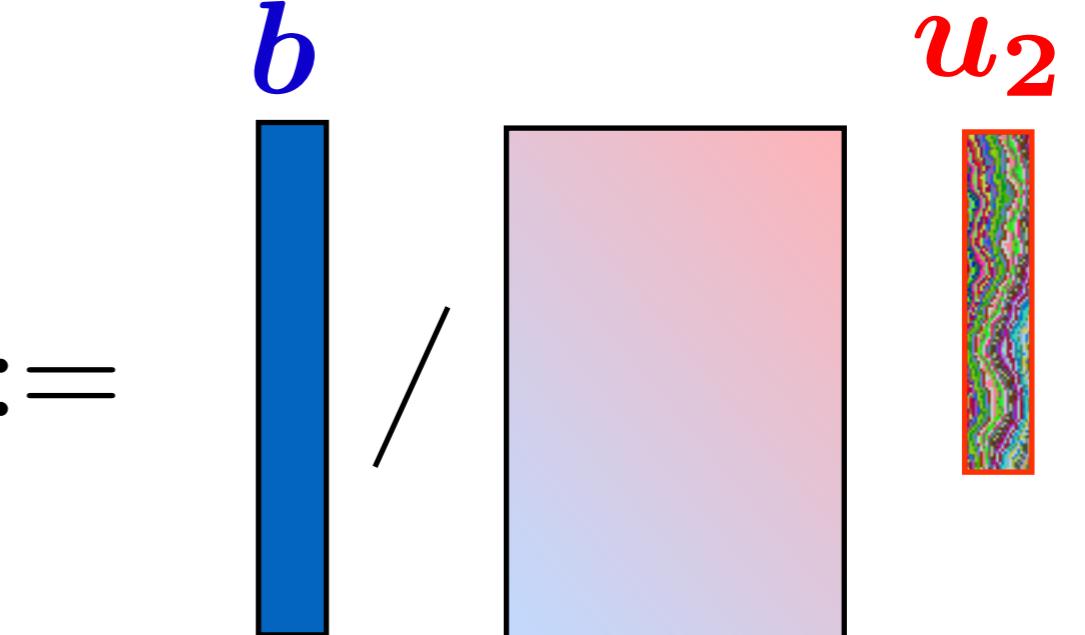
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$

$v_1$

$$\begin{matrix} \mathbf{a} & \mathbf{b} \end{matrix}$$


$$:= \begin{matrix} \mathbf{a} \\ \mathbf{b} \end{matrix} / \begin{matrix} \mathbf{K} \\ \mathbf{v}_1 \end{matrix}$$


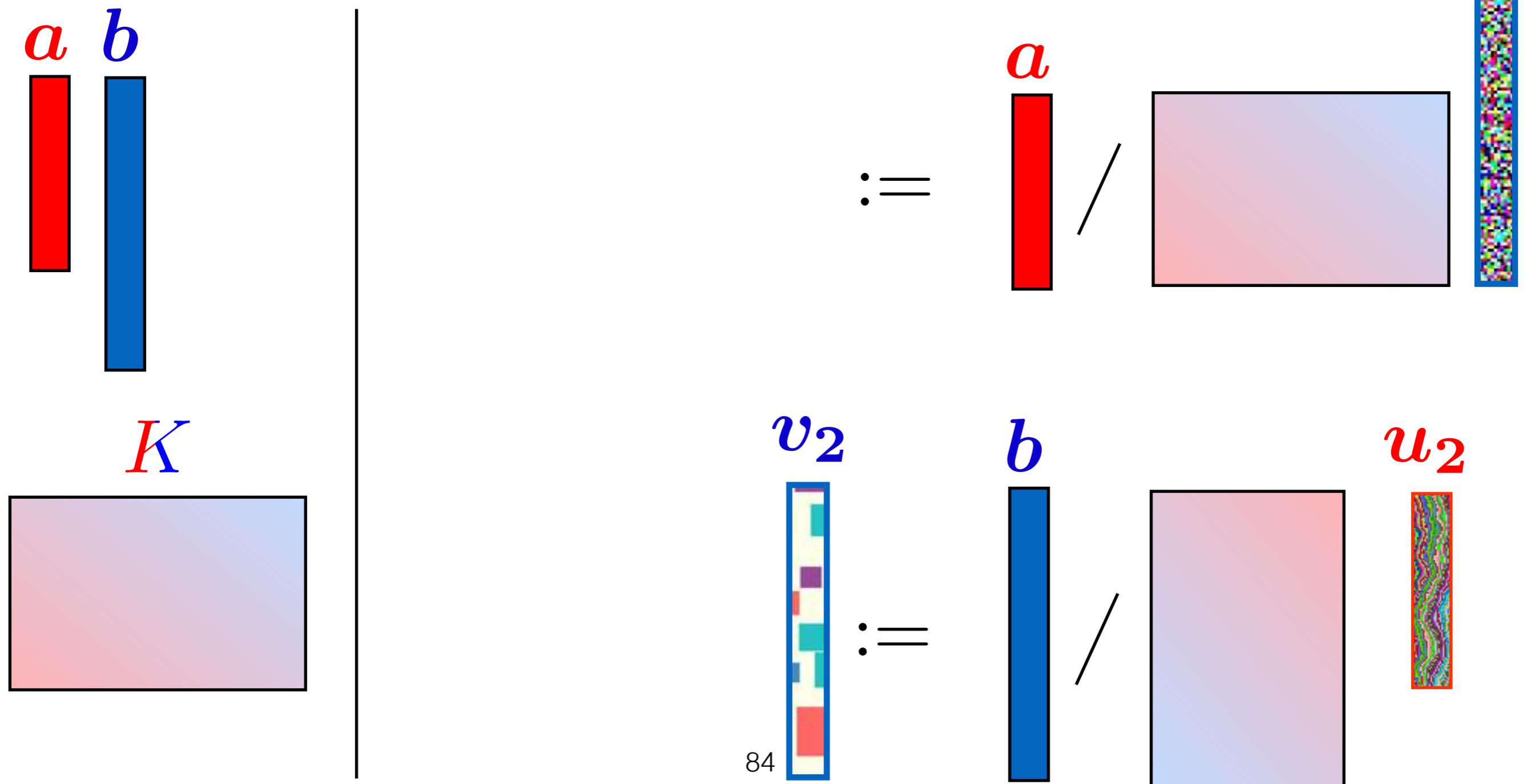
$$\mathbf{K}$$


$$:= \begin{matrix} \mathbf{b} \\ \mathbf{K} \end{matrix} / \begin{matrix} \mathbf{u}_2 \\ \mathbf{v}_1 \end{matrix}$$


# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

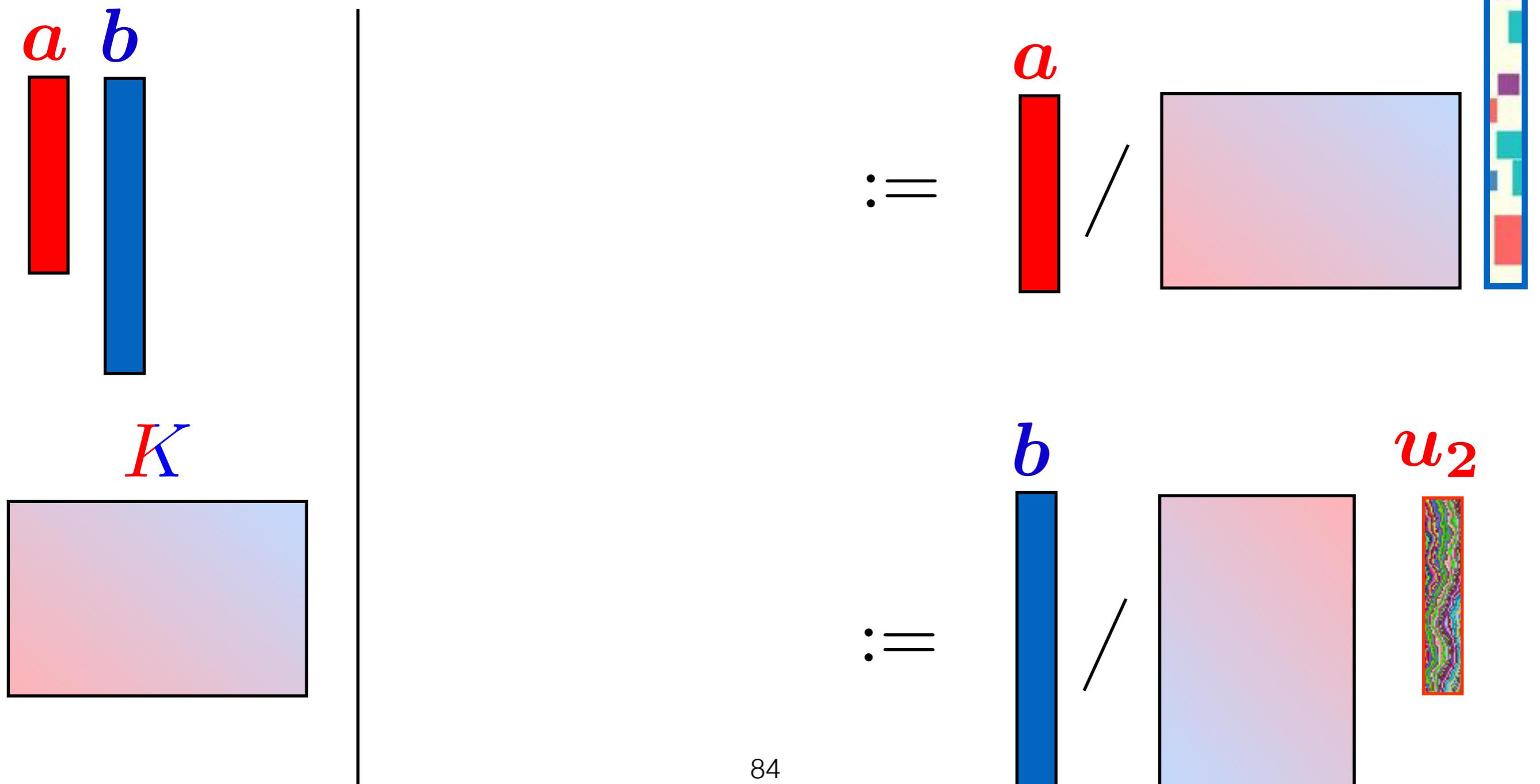
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

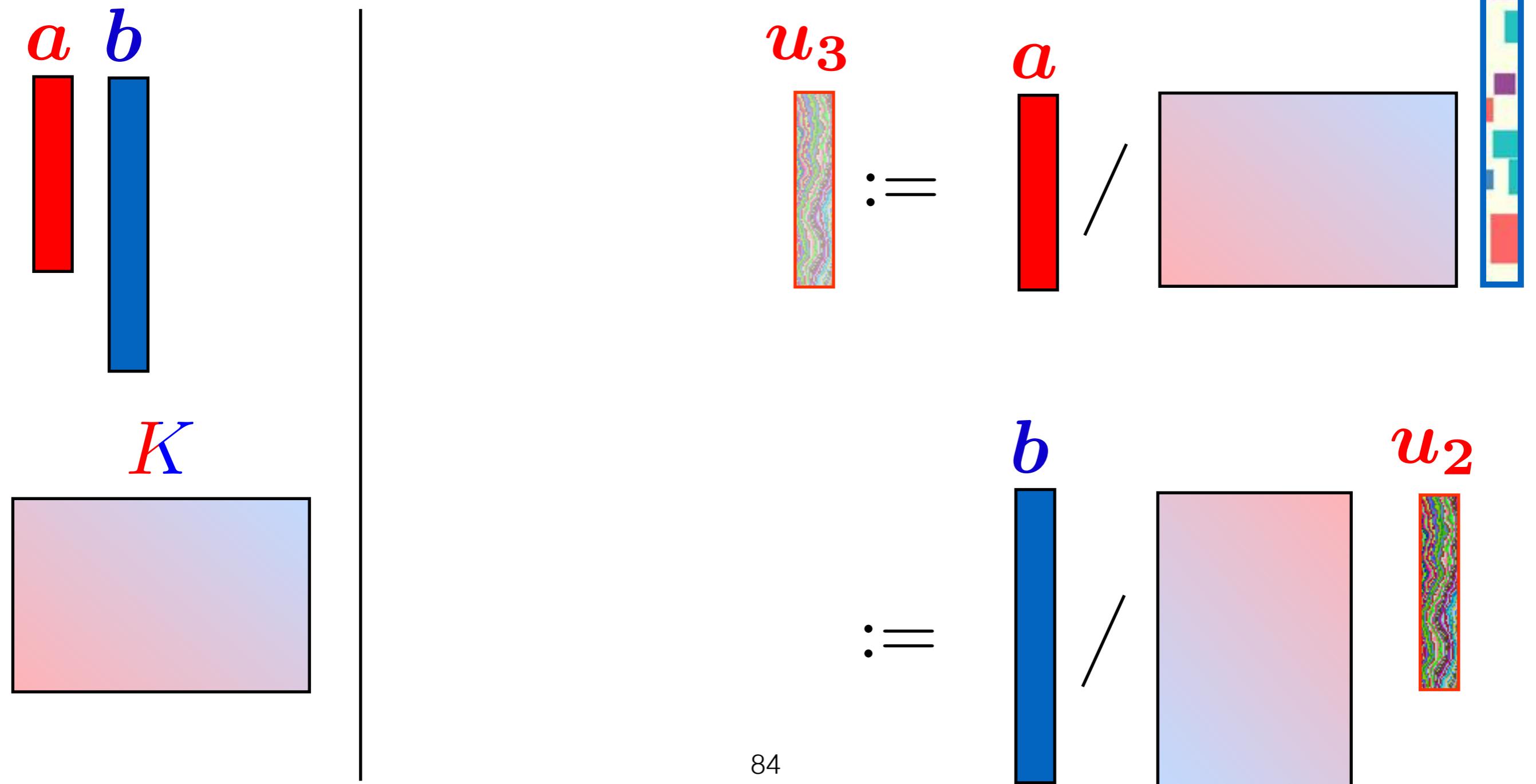
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

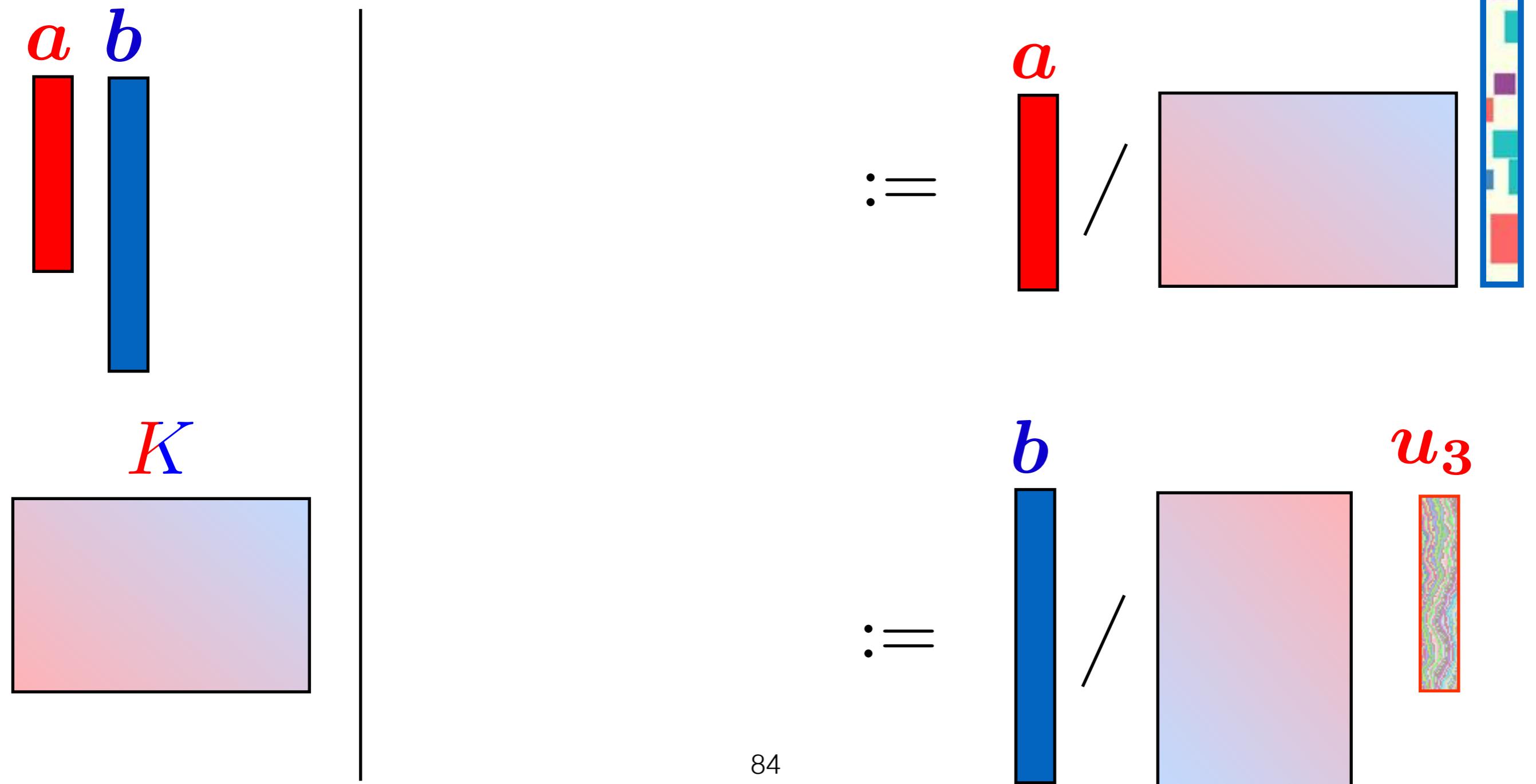
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

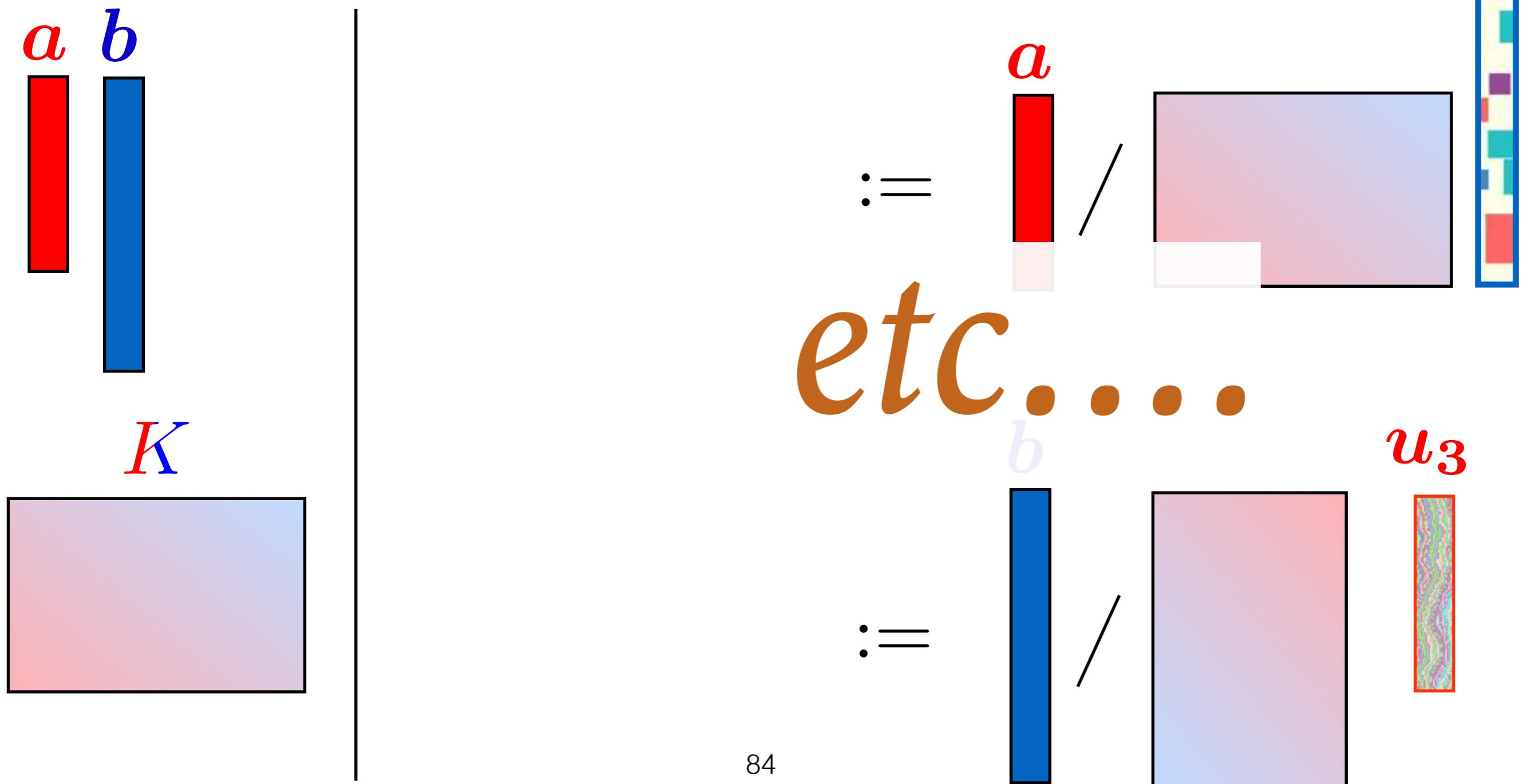
$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



# Fast & Scalable Algorithm

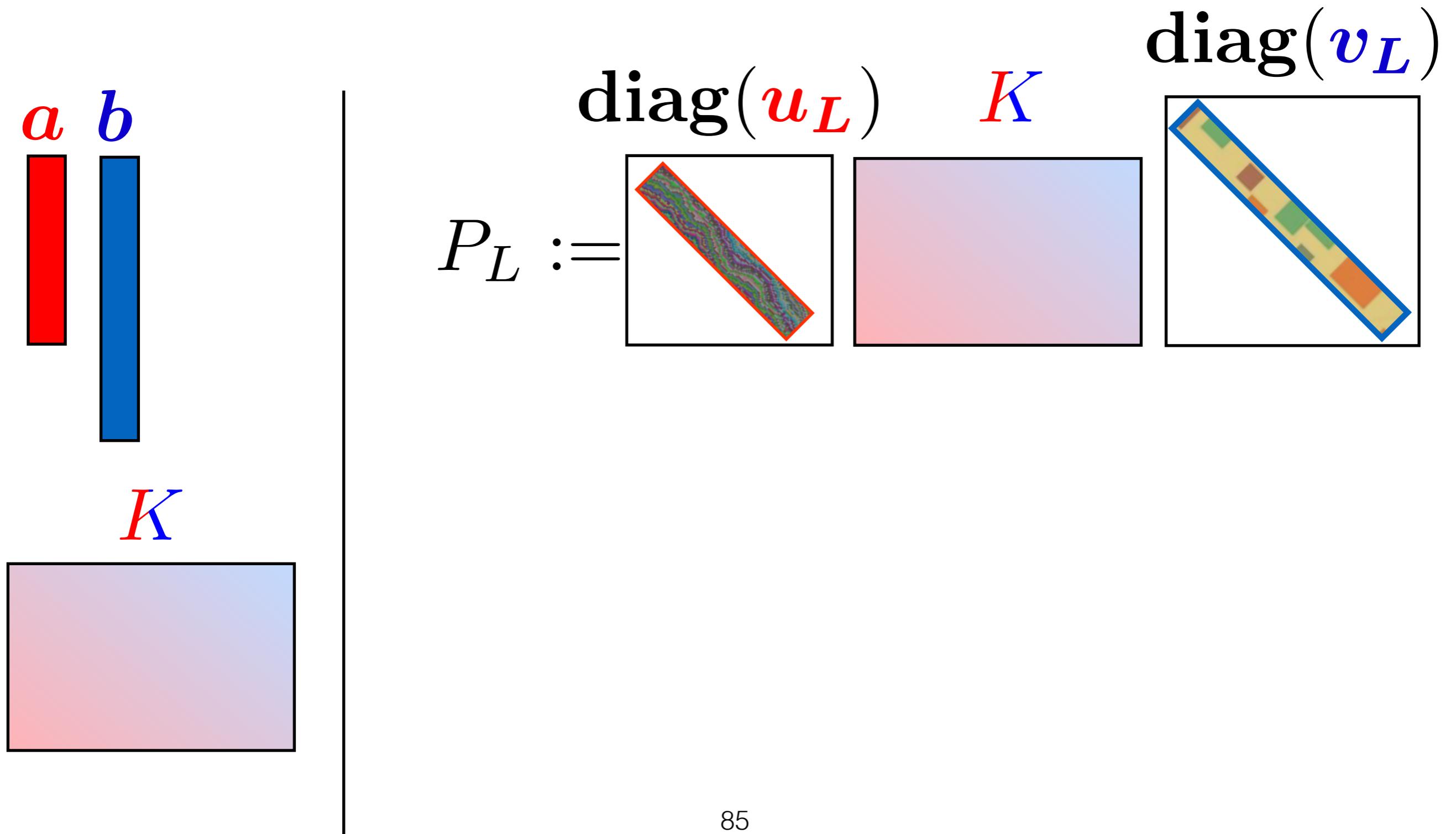
- [Sinkhorn'64] fixed-point iterations for  $(\mathbf{u}, \mathbf{v})$

$$\mathbf{u} \leftarrow \mathbf{a}/K\mathbf{v}, \quad \mathbf{v} \leftarrow \mathbf{b}/K^T \mathbf{u}$$



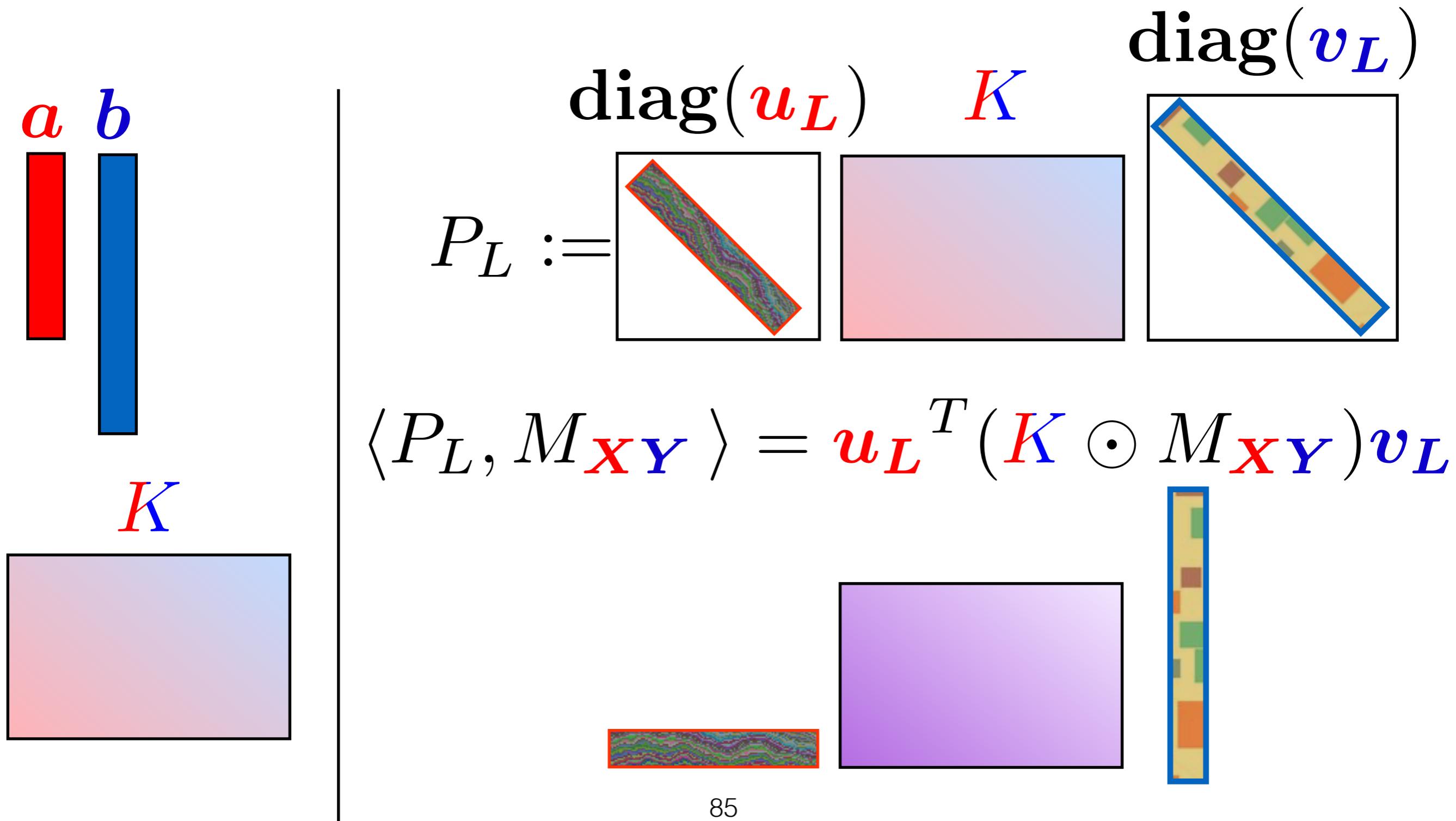
# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations.



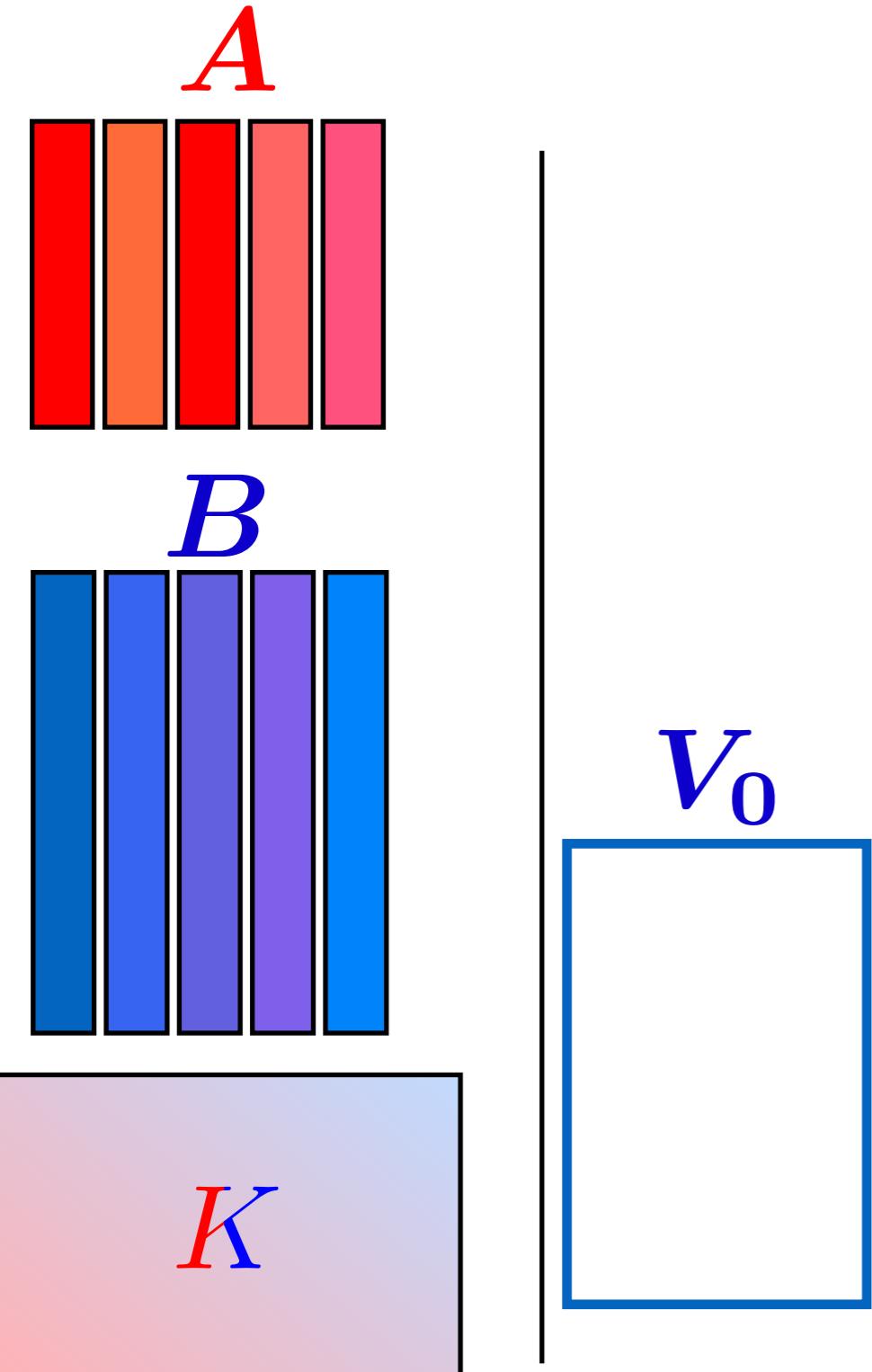
# Fast & Scalable Algorithm

- [Sinkhorn'64] fixed-point iterations.



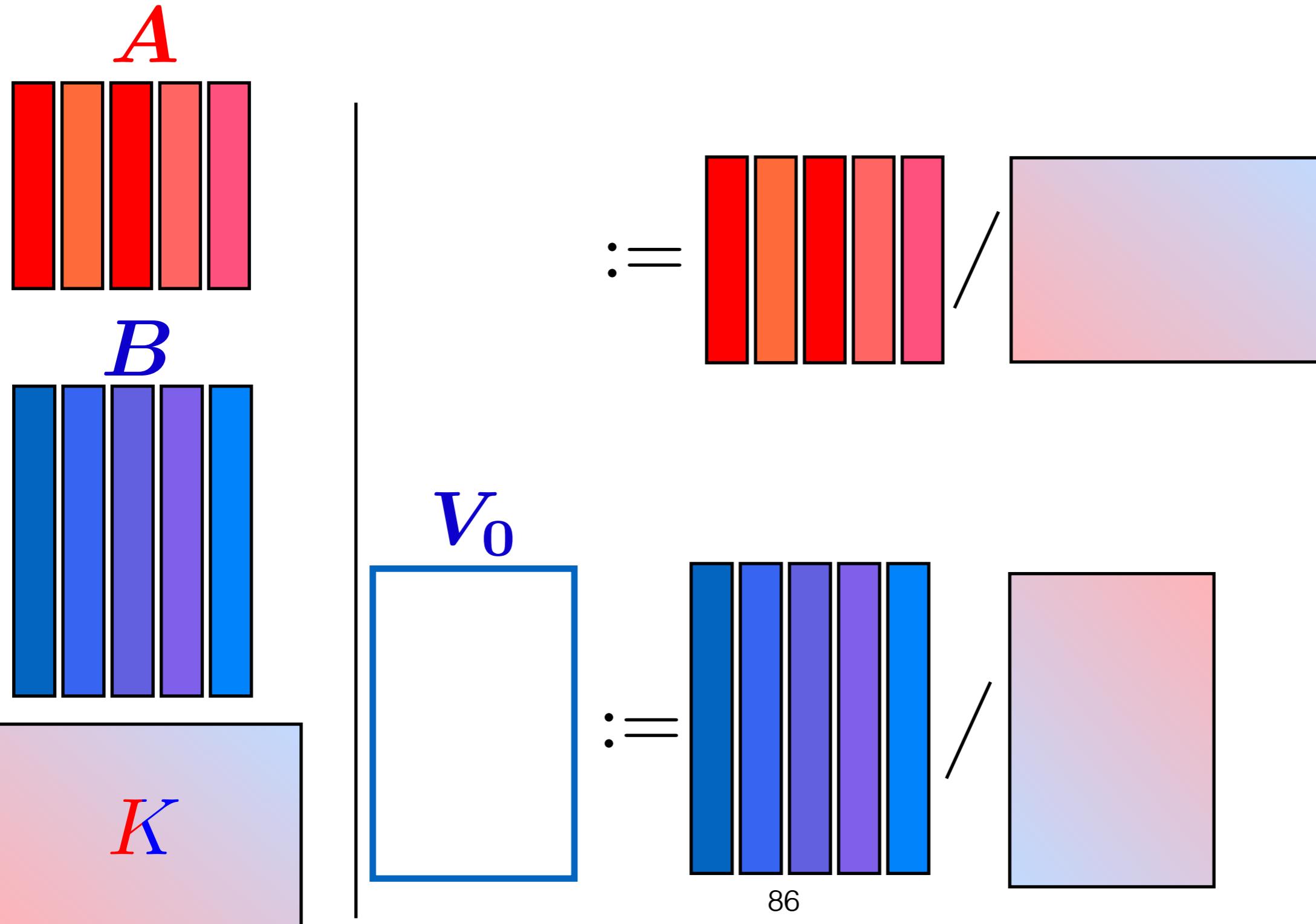
# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



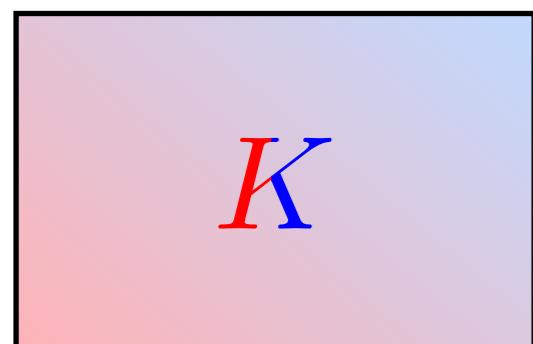
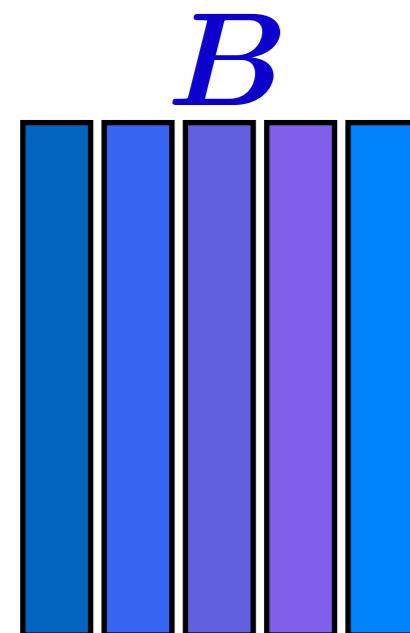
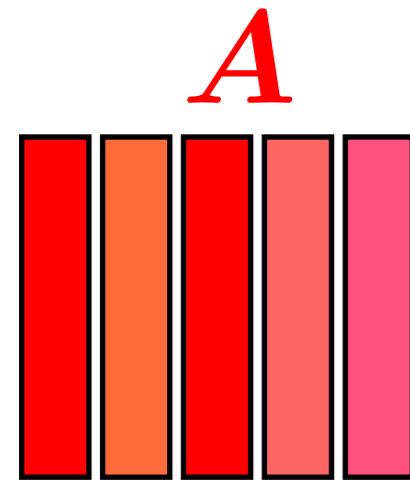
# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

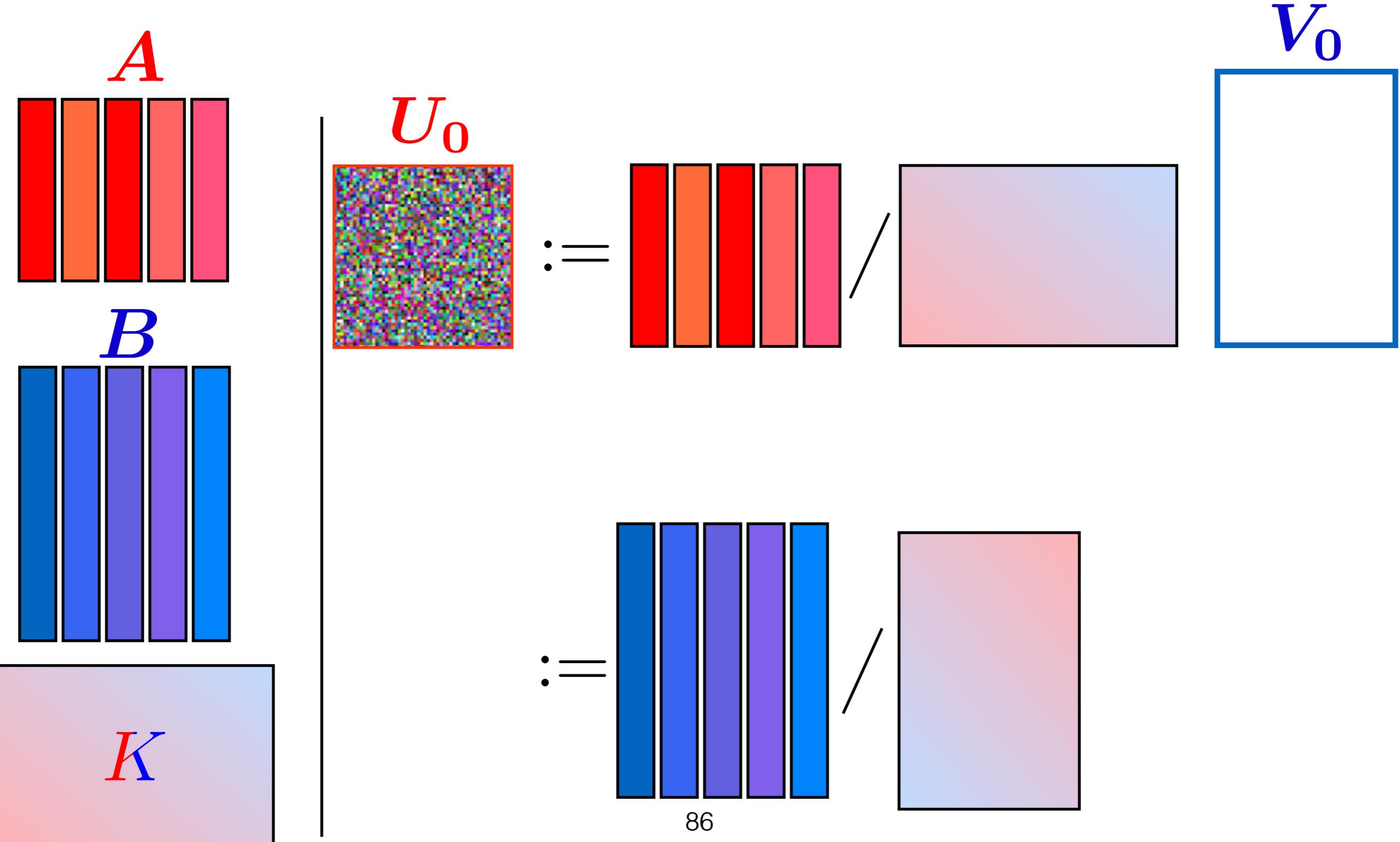
A square matrix with a color gradient from pink at the bottom-left to light blue at the top-right. To its right is the letter  $V_0$  in blue.

$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

A square matrix with a color gradient from pink at the bottom-left to light blue at the top-right. To its right is the letter  $V_0$  in blue.

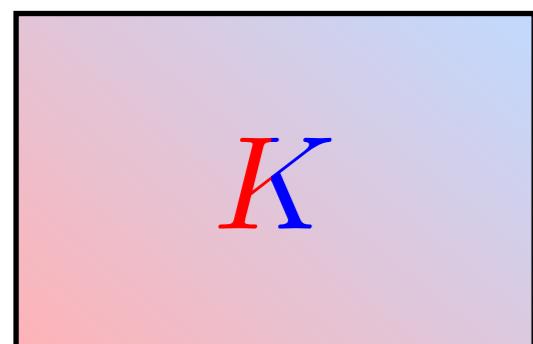
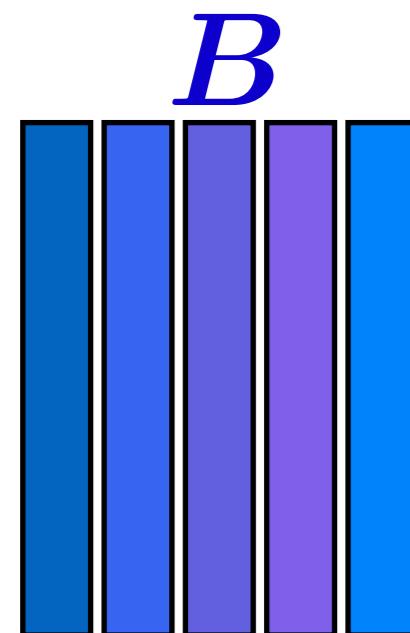
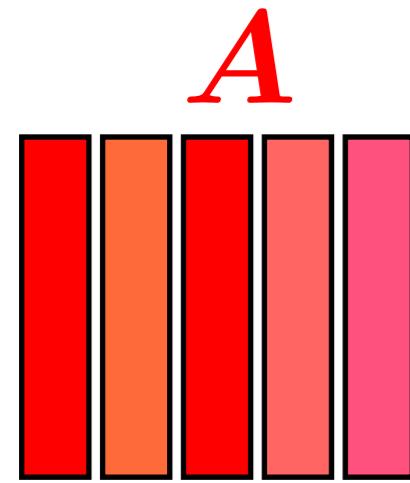
# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations

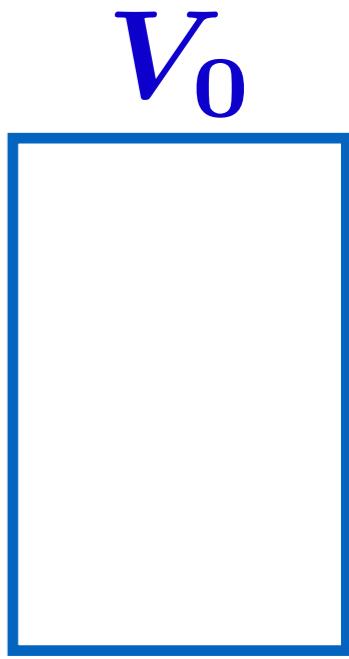
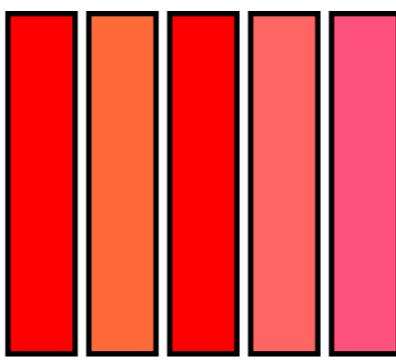


# Also embarrassingly parallel

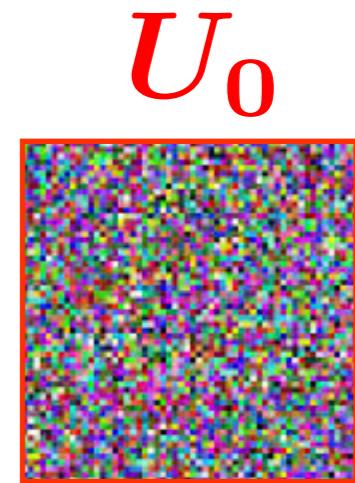
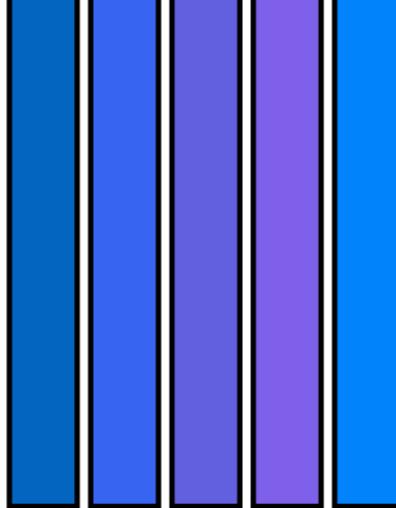
- [Sinkhorn'64] with *matrix* fixed-point iterations



$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

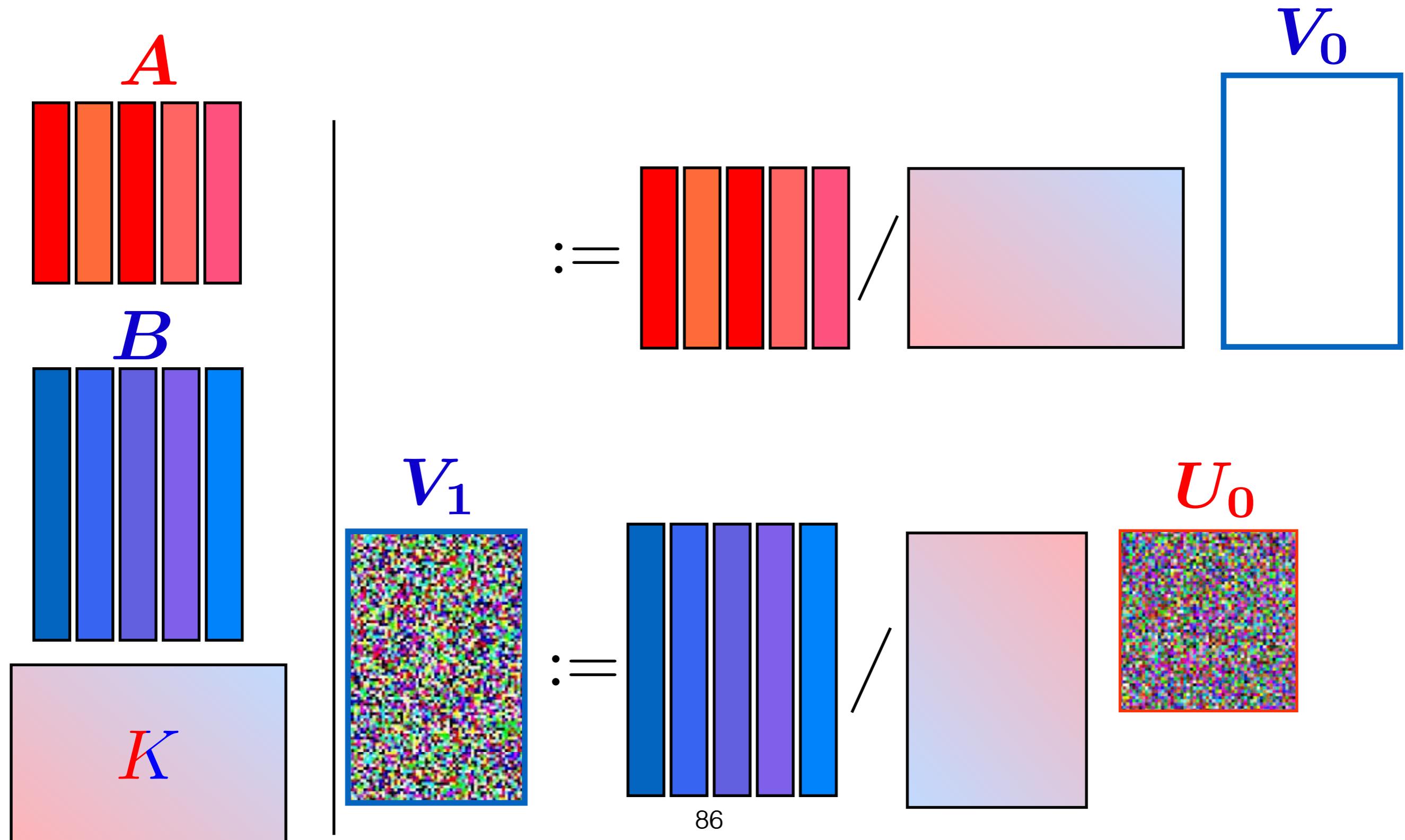


$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$



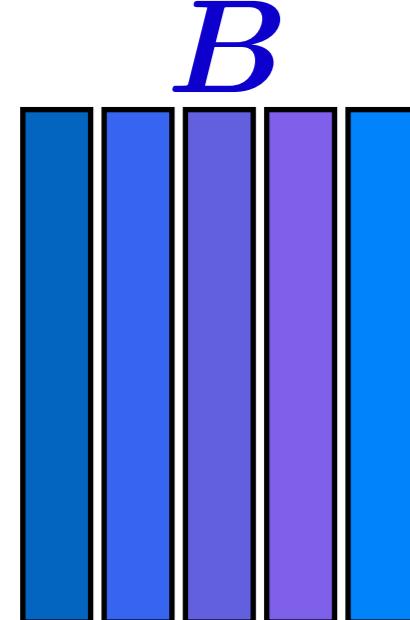
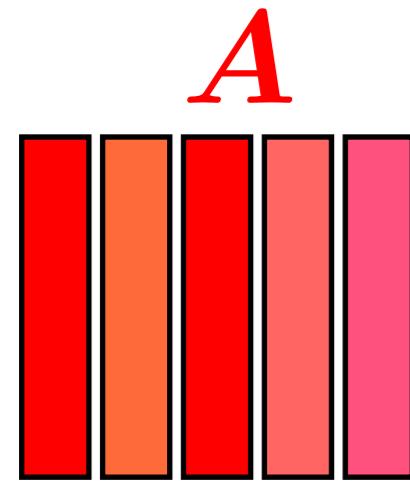
# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



$$:= \begin{array}{c} | \\ \text{red bar} \\ | \\ \text{orange bar} \\ | \\ \text{red bar} \\ | \\ \text{pink bar} \\ | \\ \text{red bar} \end{array}$$

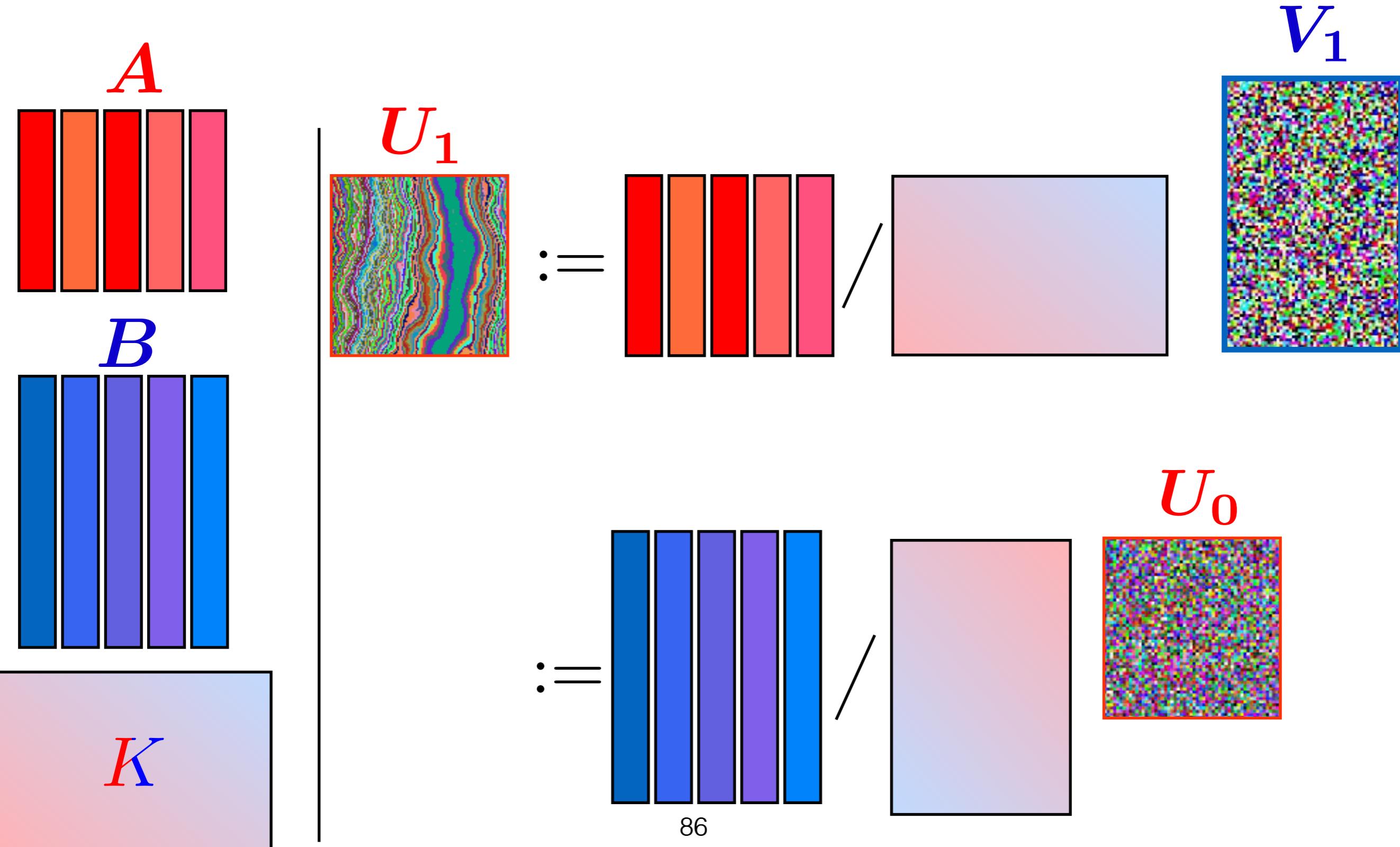


$$:= \begin{array}{c} | \\ \text{blue bar} \\ | \\ \text{dark blue bar} \\ | \\ \text{purple bar} \\ | \\ \text{light purple bar} \\ | \\ \text{blue bar} \end{array}$$



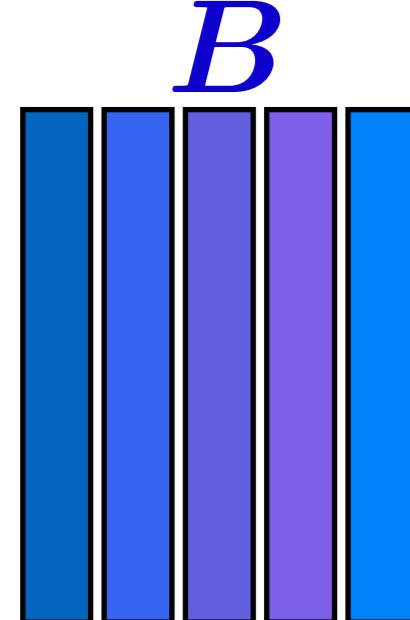
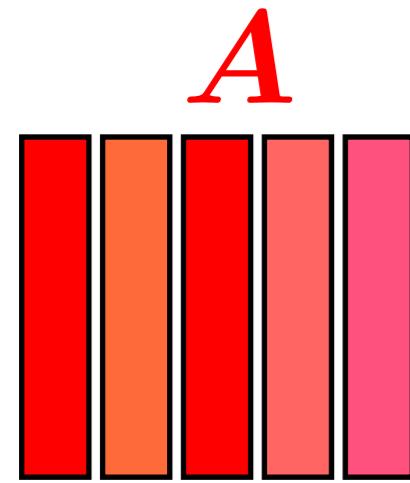
# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations

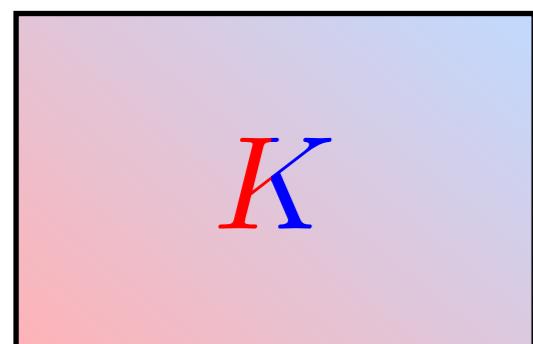
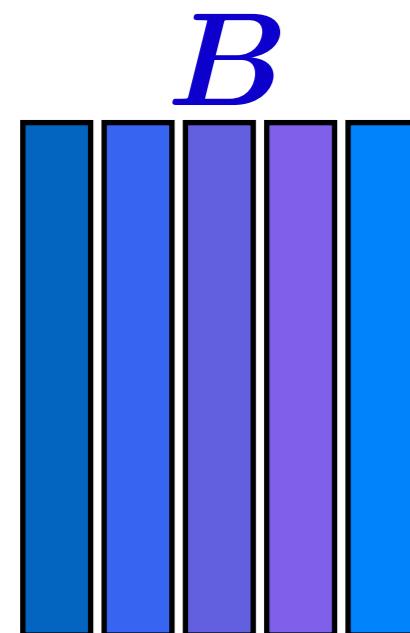
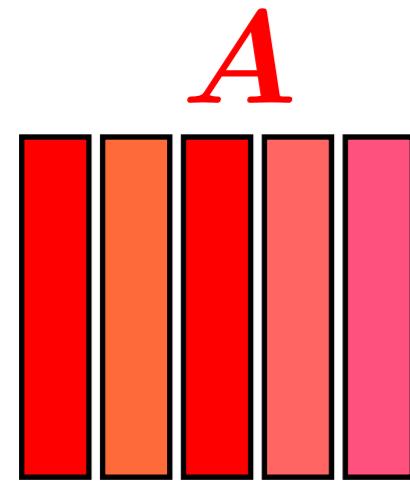


$$\begin{aligned} &:= \text{[A]} \quad \text{[V}_1\text{]} \\ &:= \text{[B]} \quad \text{[U}_1\text{]} \end{aligned}$$

The diagram illustrates the Sinkhorn fixed-point iteration process. It shows two main stages of computation. In the first stage, matrix  $A$  is multiplied by vector  $V_1$ , resulting in a vector with a smooth, linear gradient transition from red to pink. In the second stage, matrix  $B$  is multiplied by vector  $U_1$ , resulting in a vector with a smooth, linear gradient transition from blue to purple. The labels  $A$ ,  $B$ ,  $V_1$ , and  $U_1$  are in red, while  $K$  and the intermediate results are in blue/purple.

# Also embarrassingly parallel

- [Sinkhorn'64] with *matrix* fixed-point iterations



$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

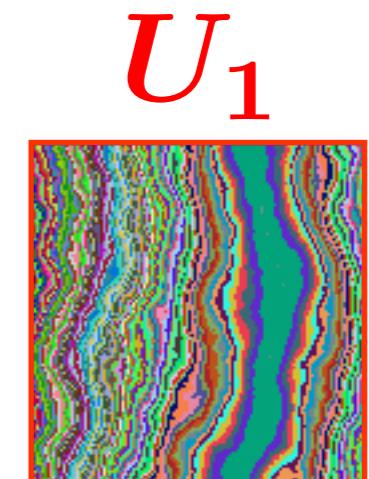
etc. . . .

$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

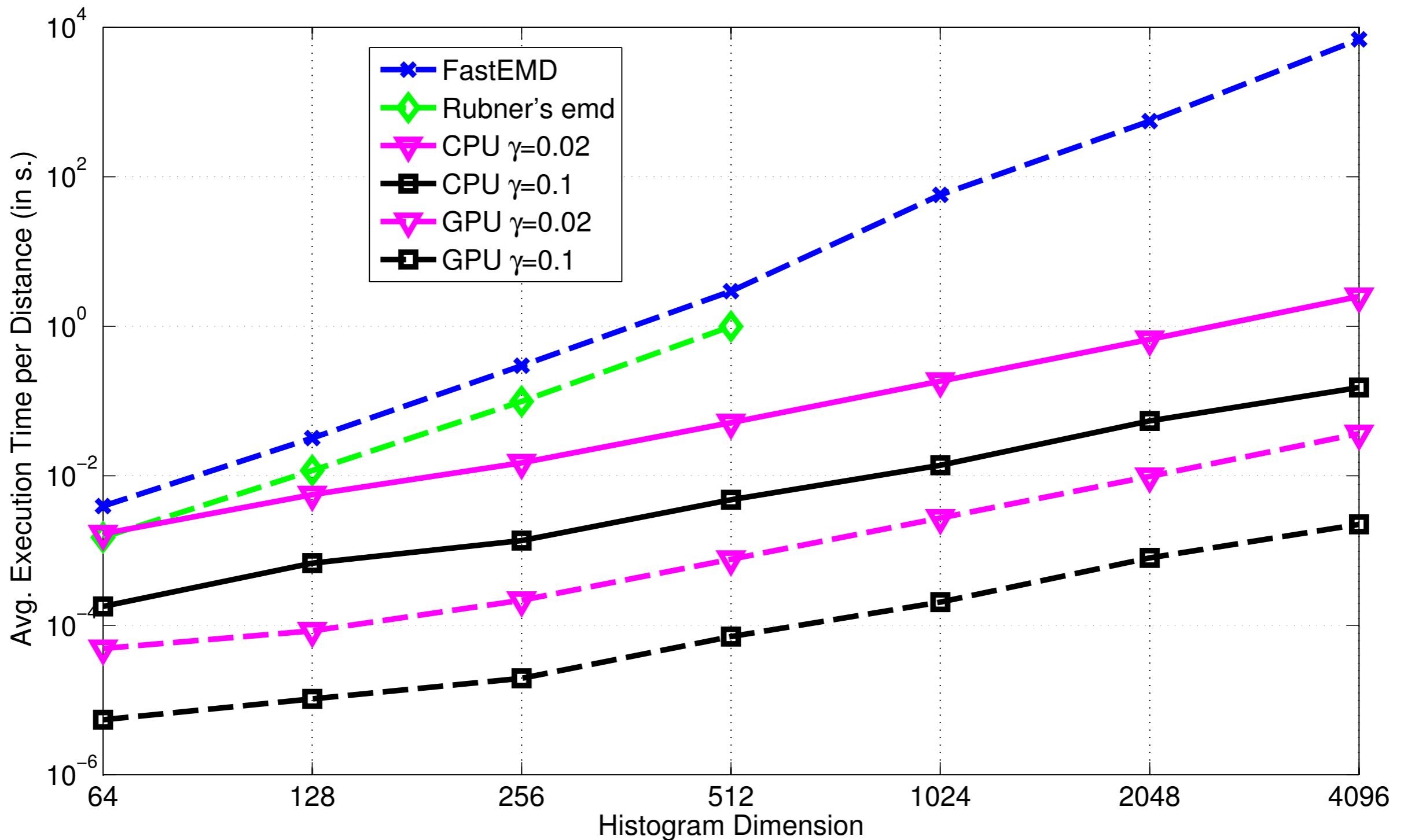
etc. . . .

$$:= \begin{array}{c} | \\ | \\ | \\ | \\ | \end{array}$$

etc. . . .



# Very Fast EMD Approx. Solver

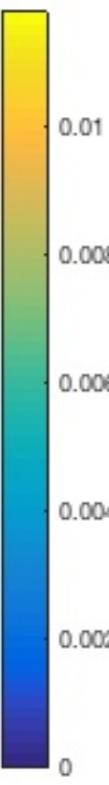
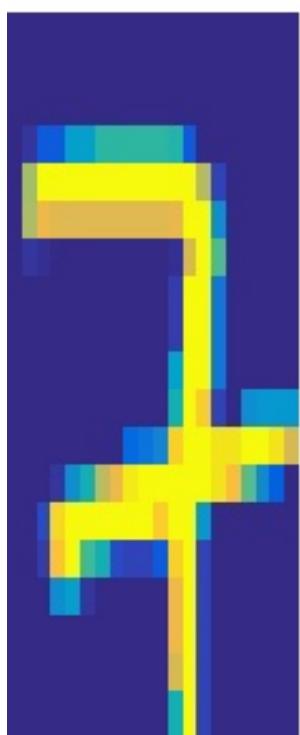


**Note.**  $(\Omega, \mathcal{D})$  is a random graph with shortest path metric, histograms sampled uniformly on simplex, Sinkhorn tolerance  $10^{-2}$ .

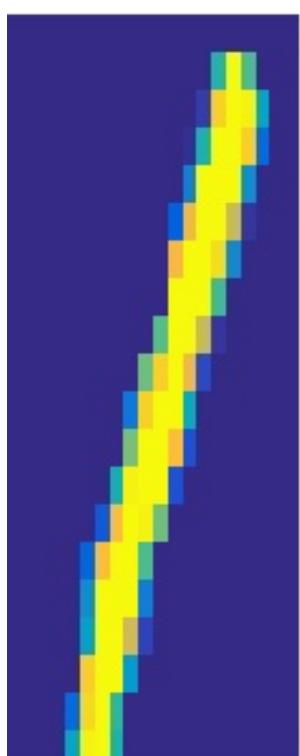
# Very Fast EMD Approx. Solver

# Very Fast EMD Approx. Solver

*a*

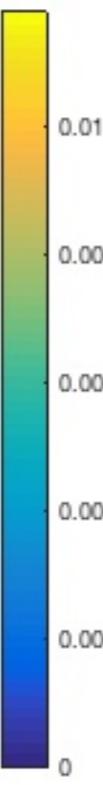
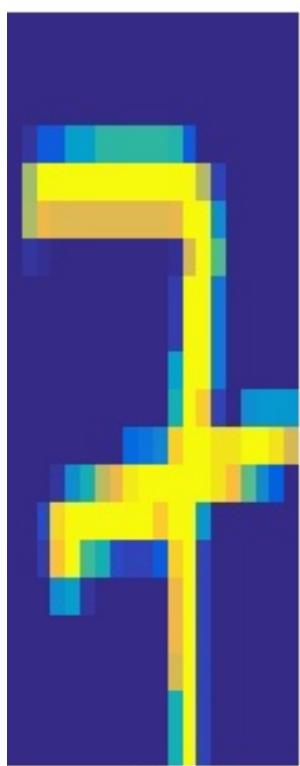


*b*

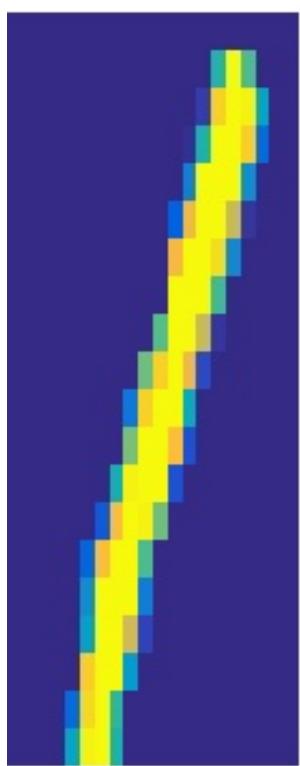


# Very Fast EMD Approx. Solver

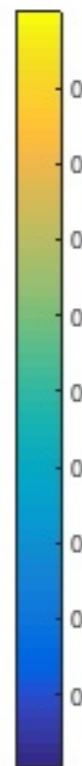
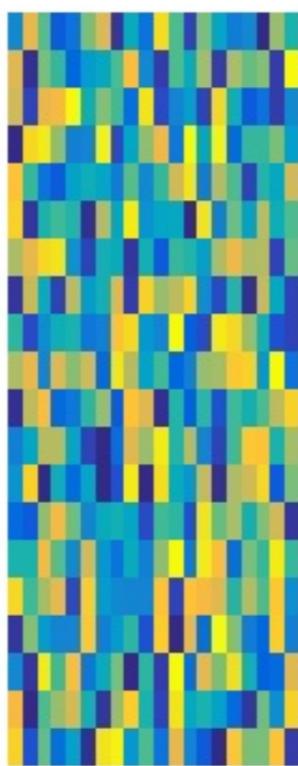
*a*



*b*

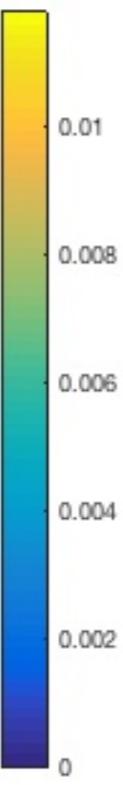
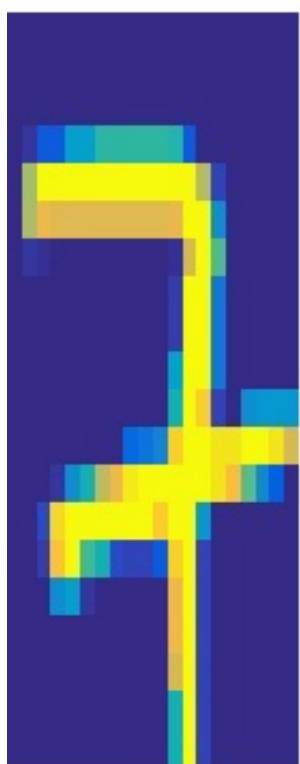


$v_1 \leftarrow \text{noise}$



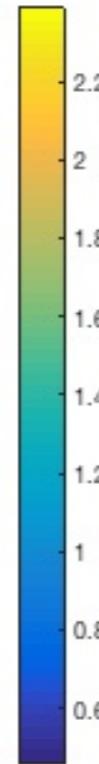
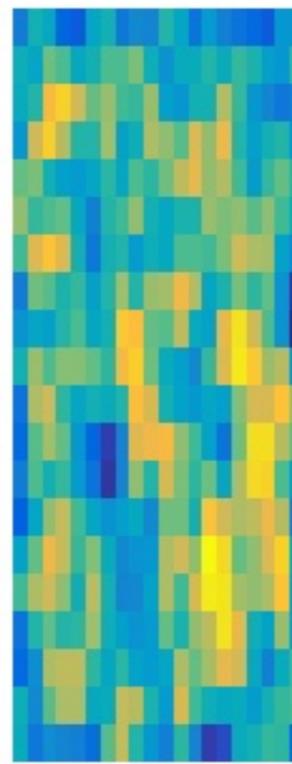
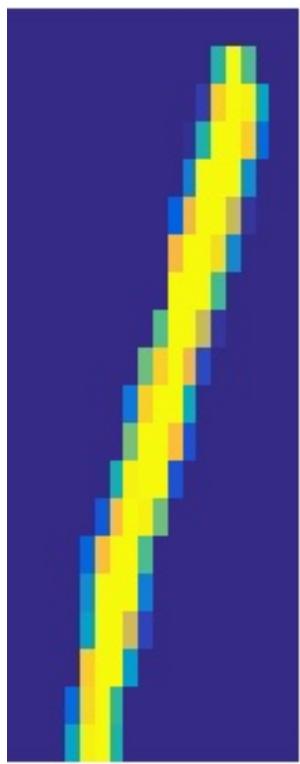
# Very Fast EMD Approx. Solver

*a*



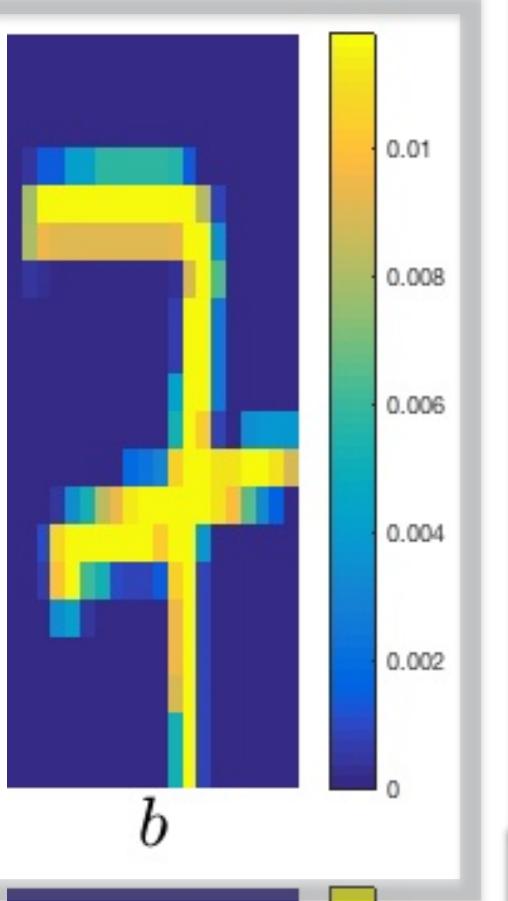
*b*

$Kv_1$

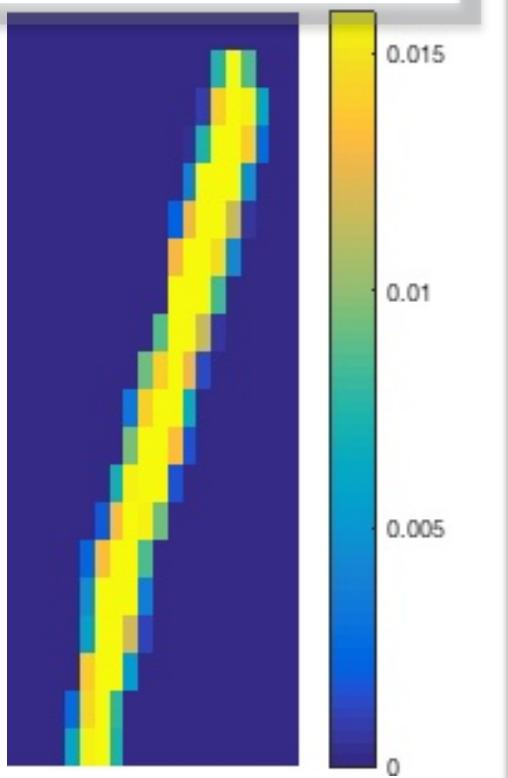


# Very Fast EMD Approx. Solver

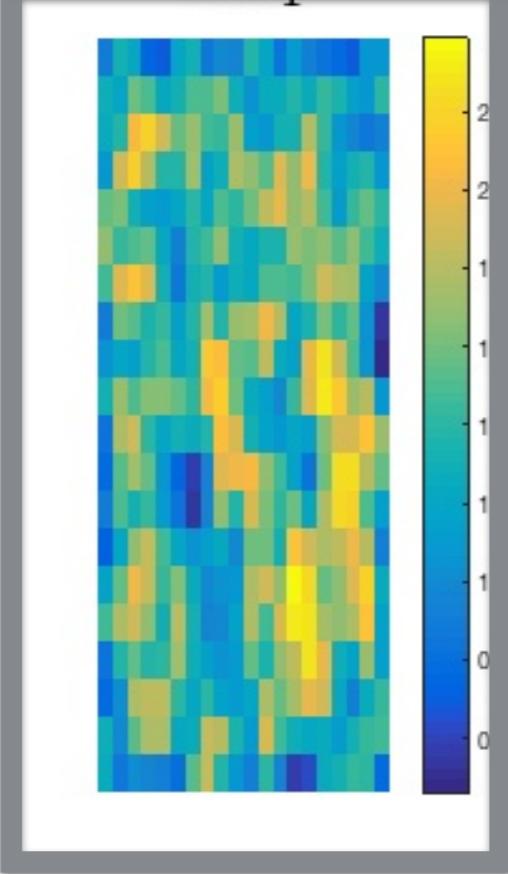
*a*



*b*

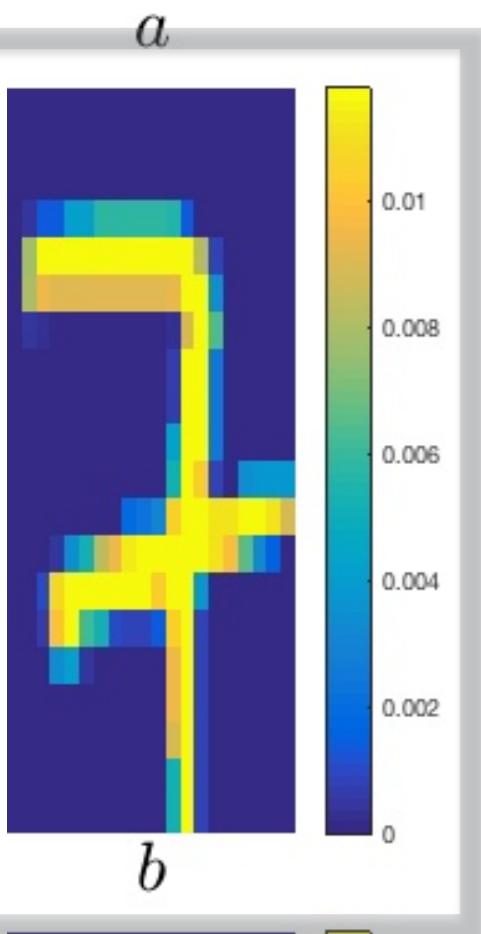


$K_{\alpha\beta}$

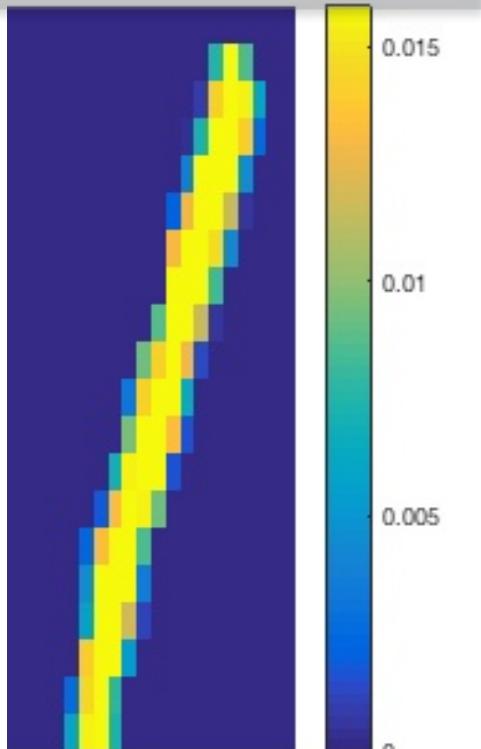


# Very Fast EMD Approx. Solver

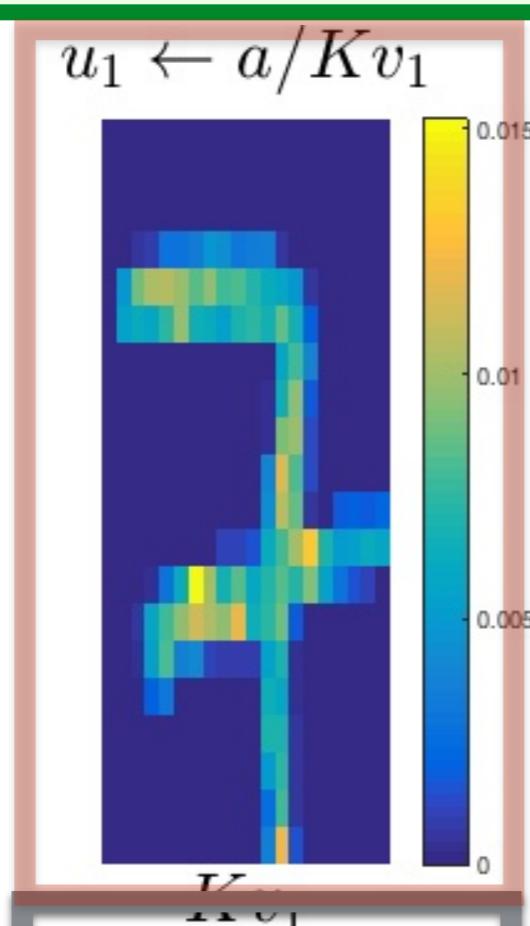
*a*



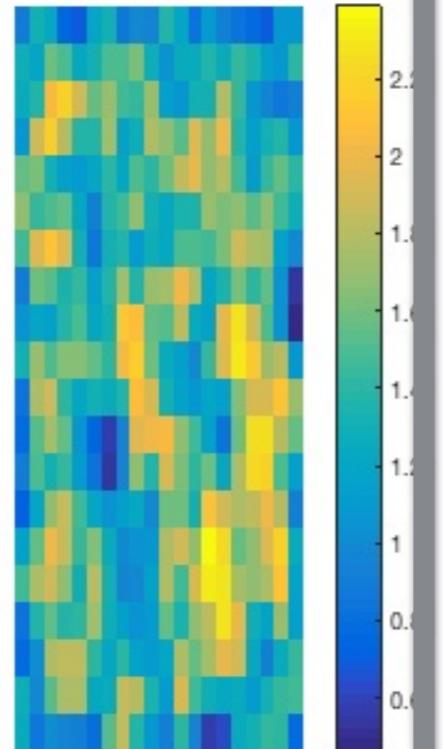
*b*



$u_1 \leftarrow a/Kv_1$

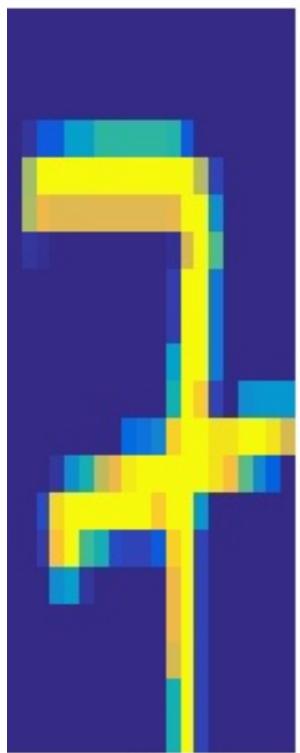


$Kv_1$

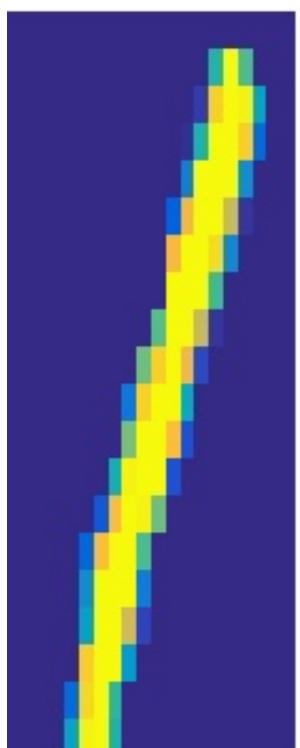


# Very Fast EMD Approx. Solver

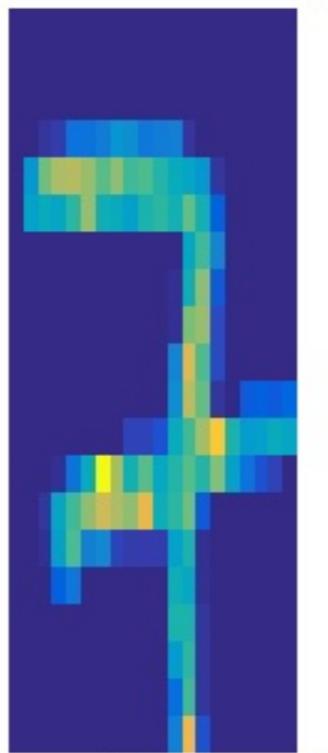
$a$



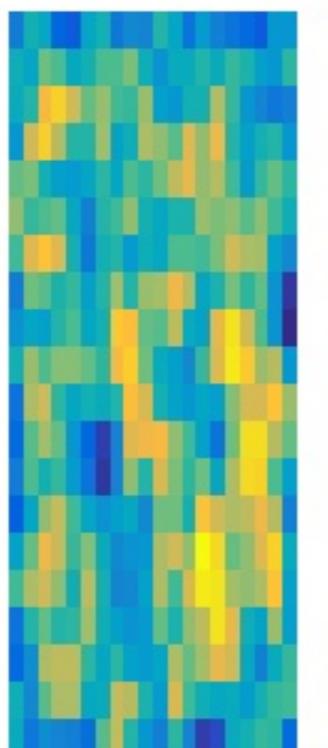
$b$



$u_1 \leftarrow a/Kv_1$



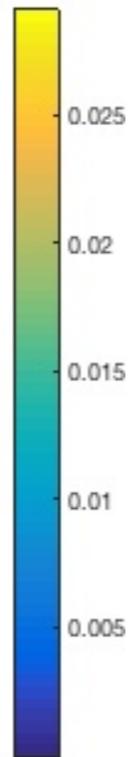
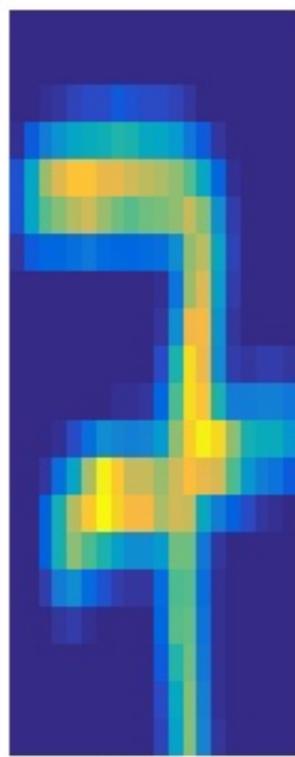
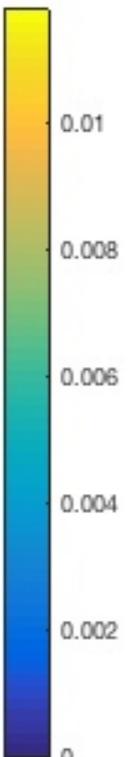
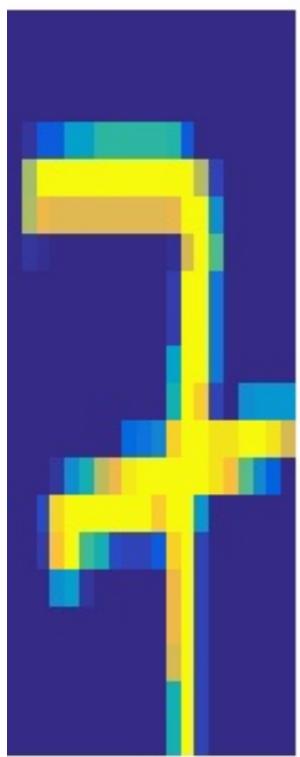
$Kv_1$



# Very Fast EMD Approx. Solver

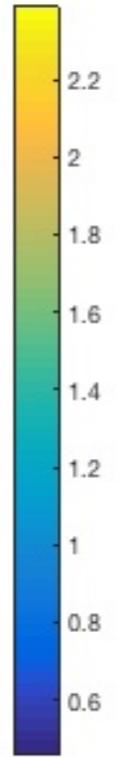
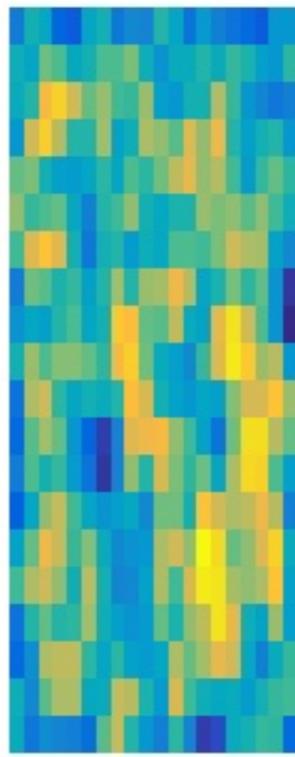
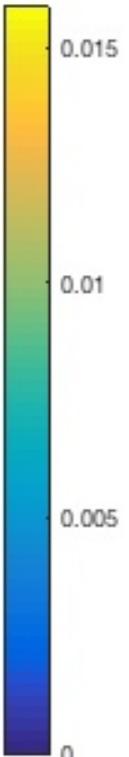
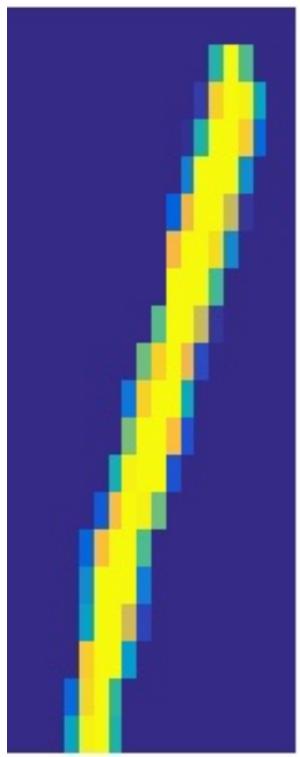
*a*

$Ku_1$



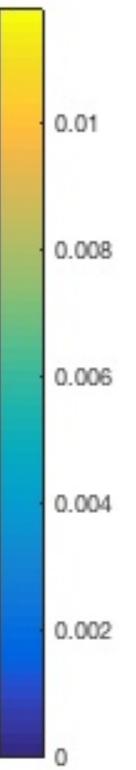
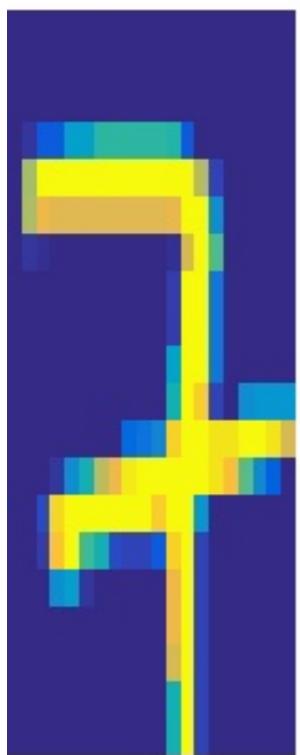
*b*

$Kv_1$

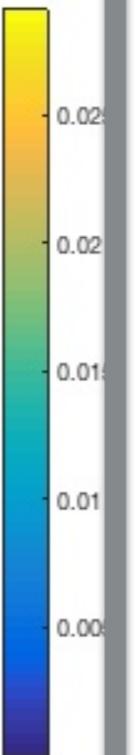
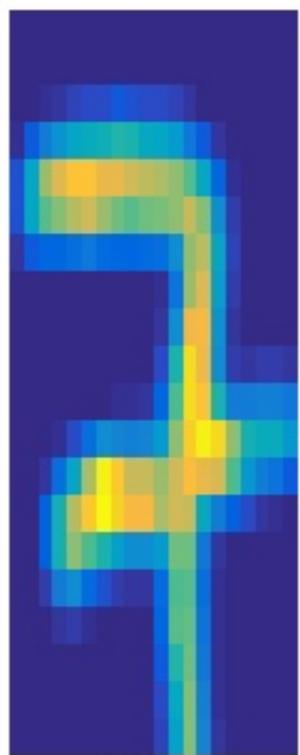


# Very Fast EMD Approx. Solver

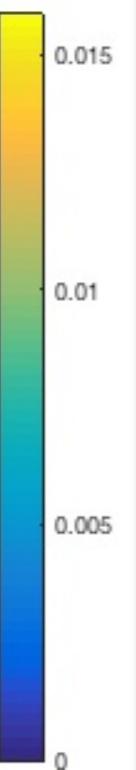
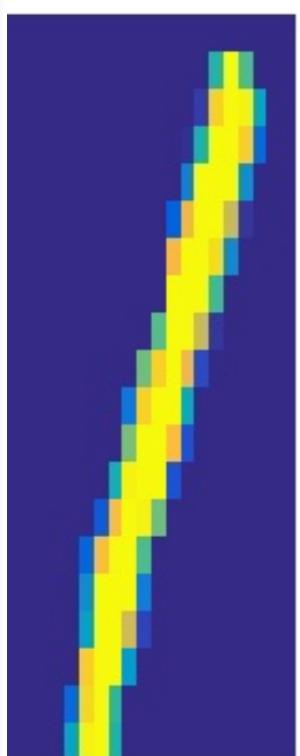
*a*



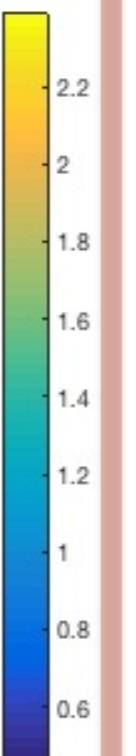
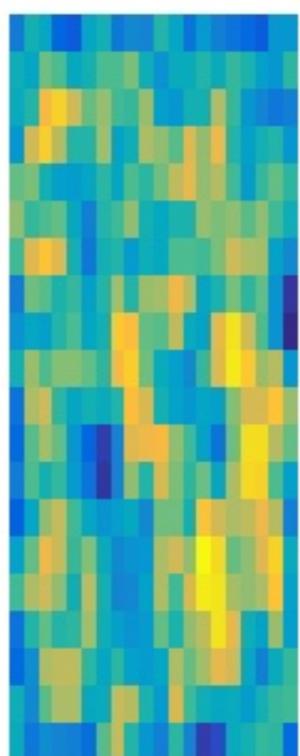
$K u_1$



*b*

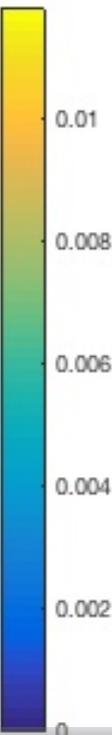
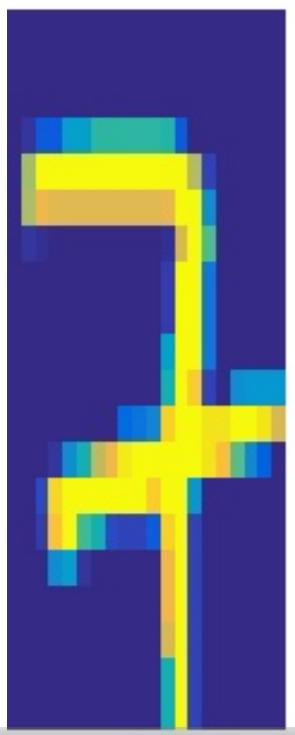


$K u_1$

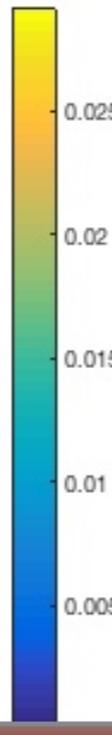
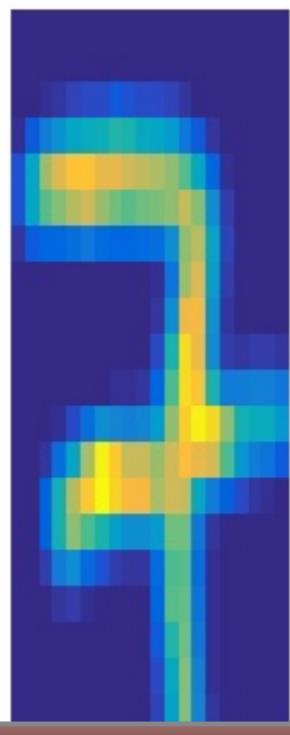


# Very Fast EMD Approx. Solver

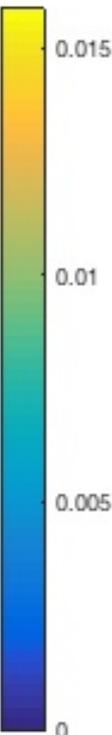
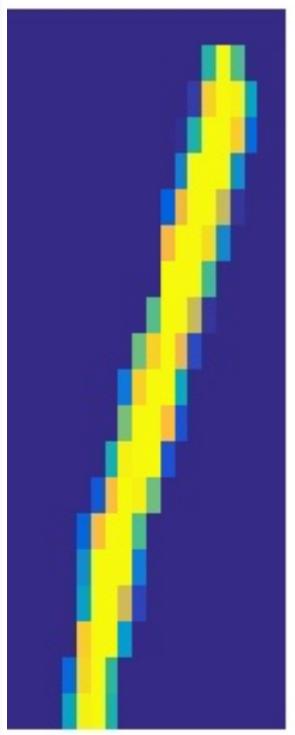
*a*



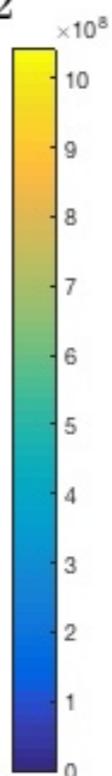
*Ku*<sub>1</sub>



*b*



*v*<sub>2</sub>  $\leftarrow$  *b*/*Ku*<sub>2</sub>



$$P_1 = D(u_1)KD(v_1)$$

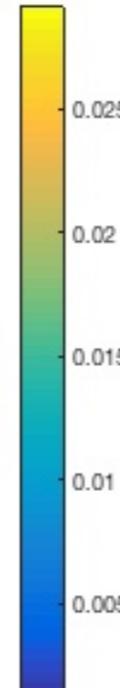
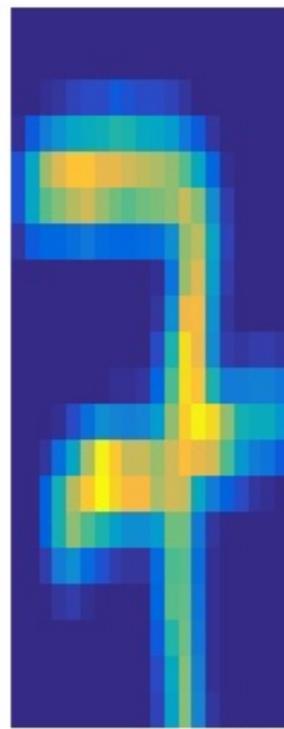
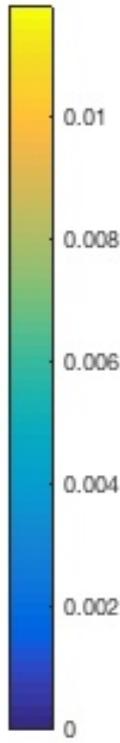
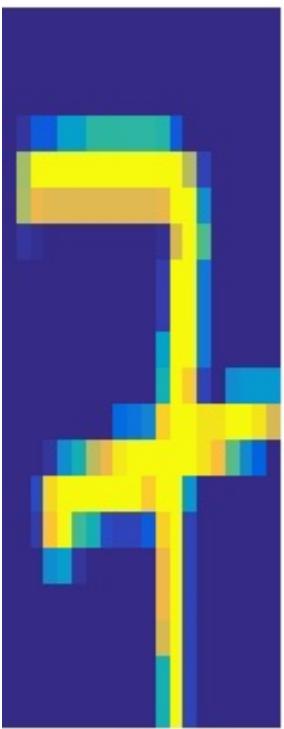
$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$



# Very Fast EMD Approx. Solver

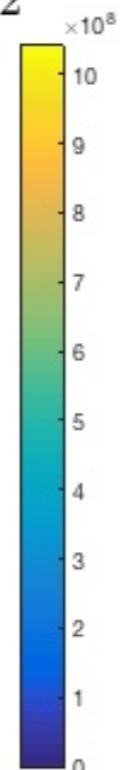
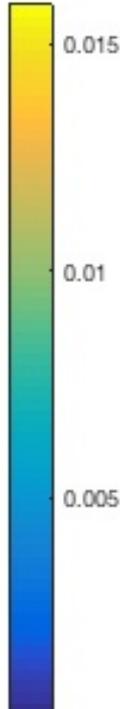
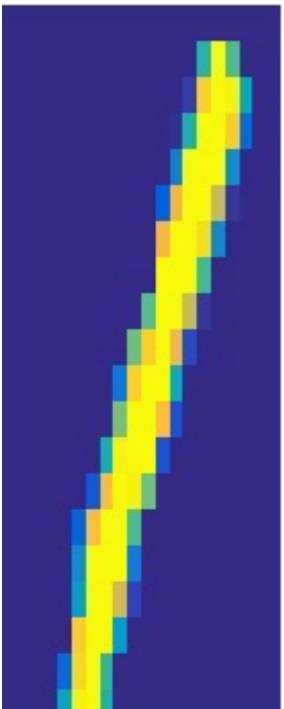
*a*

*Ku*<sub>1</sub>



*b*

*v*<sub>2</sub> ← *b*/*Ku*<sub>2</sub>



$$P_1 = D(u_1)KD(v_1)$$

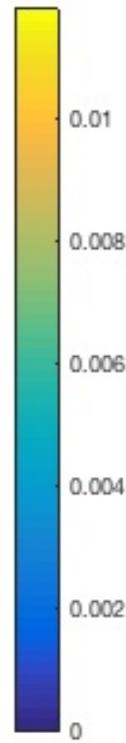
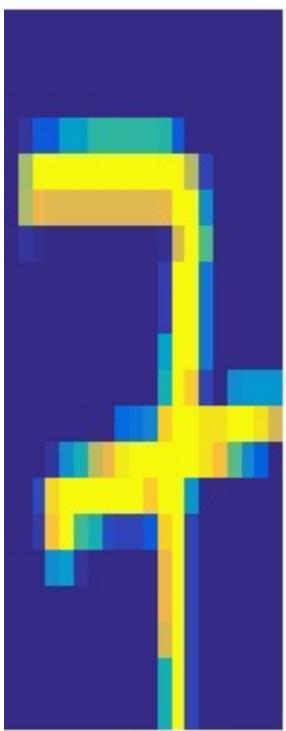
$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$



# Very Fast EMD Approx. Solver

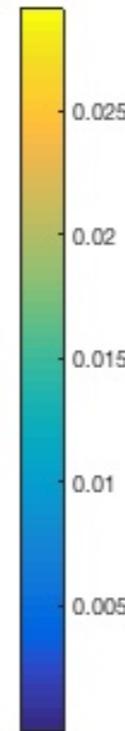
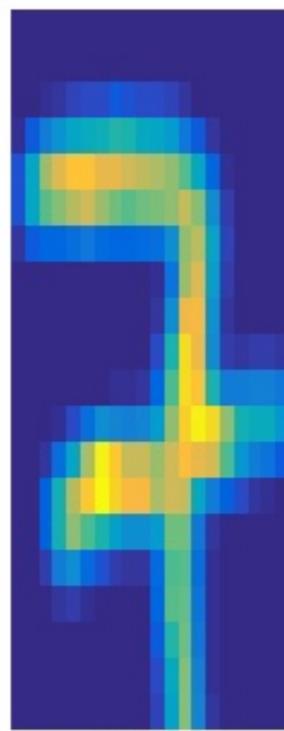
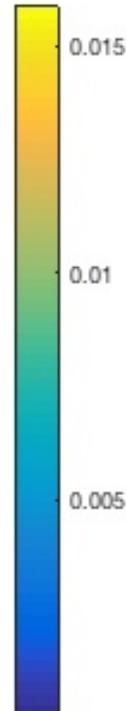
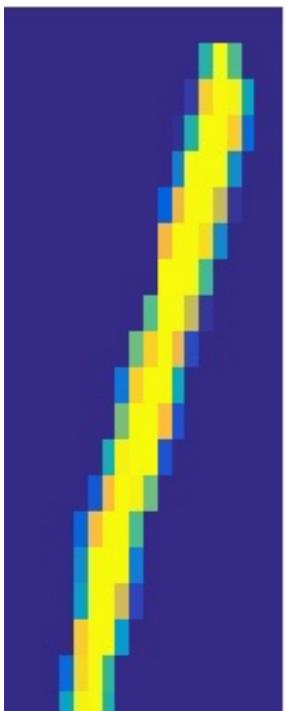
*a*

*Ku*<sub>1</sub>



*b*

*Kv*<sub>2</sub>



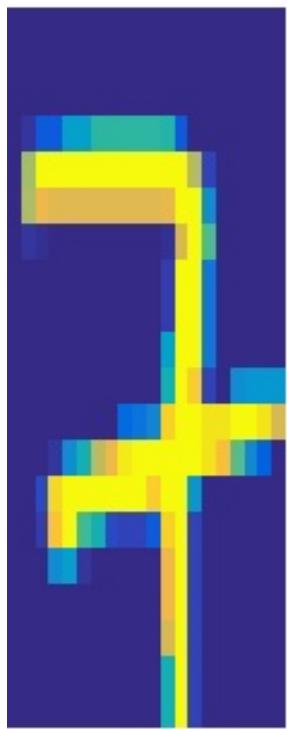
$$P_1 = D(u_1)KD(v_1)$$

$$\|P_1 - a\|_1 + \|P^T 1 - b\|_1 = 1.2691$$



# Very Fast EMD Approx. Solver

*a*



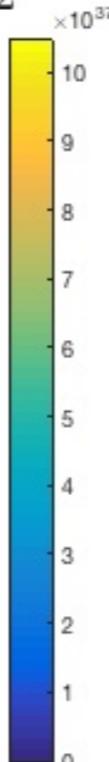
*b*



$$u_2 \leftarrow a/Kv_2$$

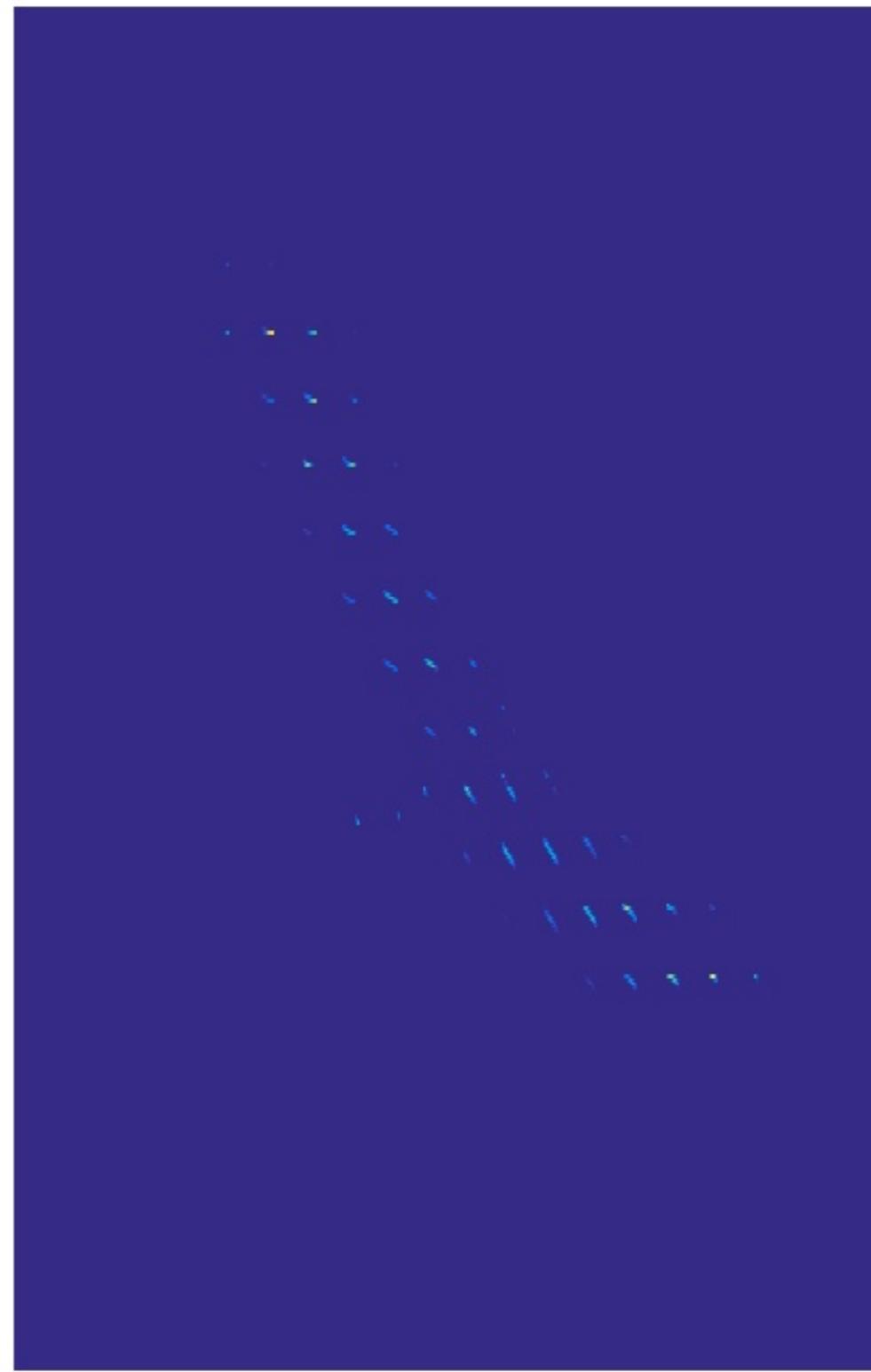


$Kv_2$



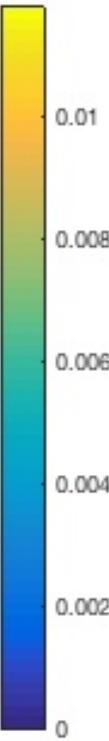
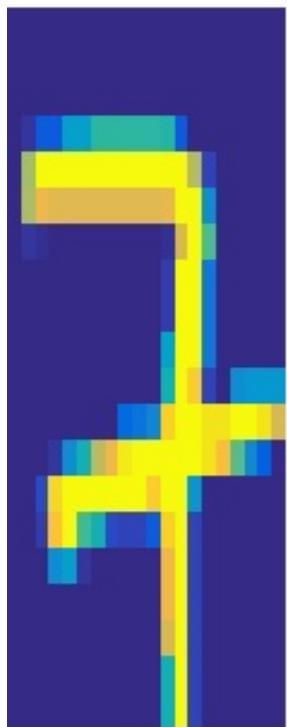
$$P_1 = D(u_1)KD(v_1)$$

$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$

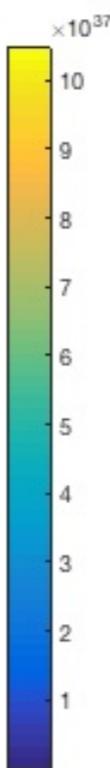


# Very Fast EMD Approx. Solver

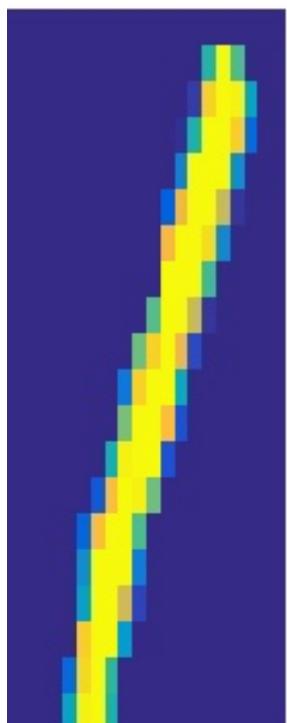
*a*



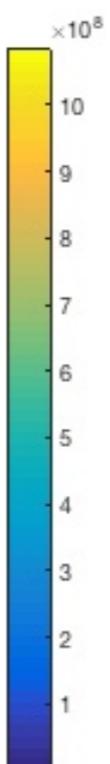
$Ku_2$



*b*

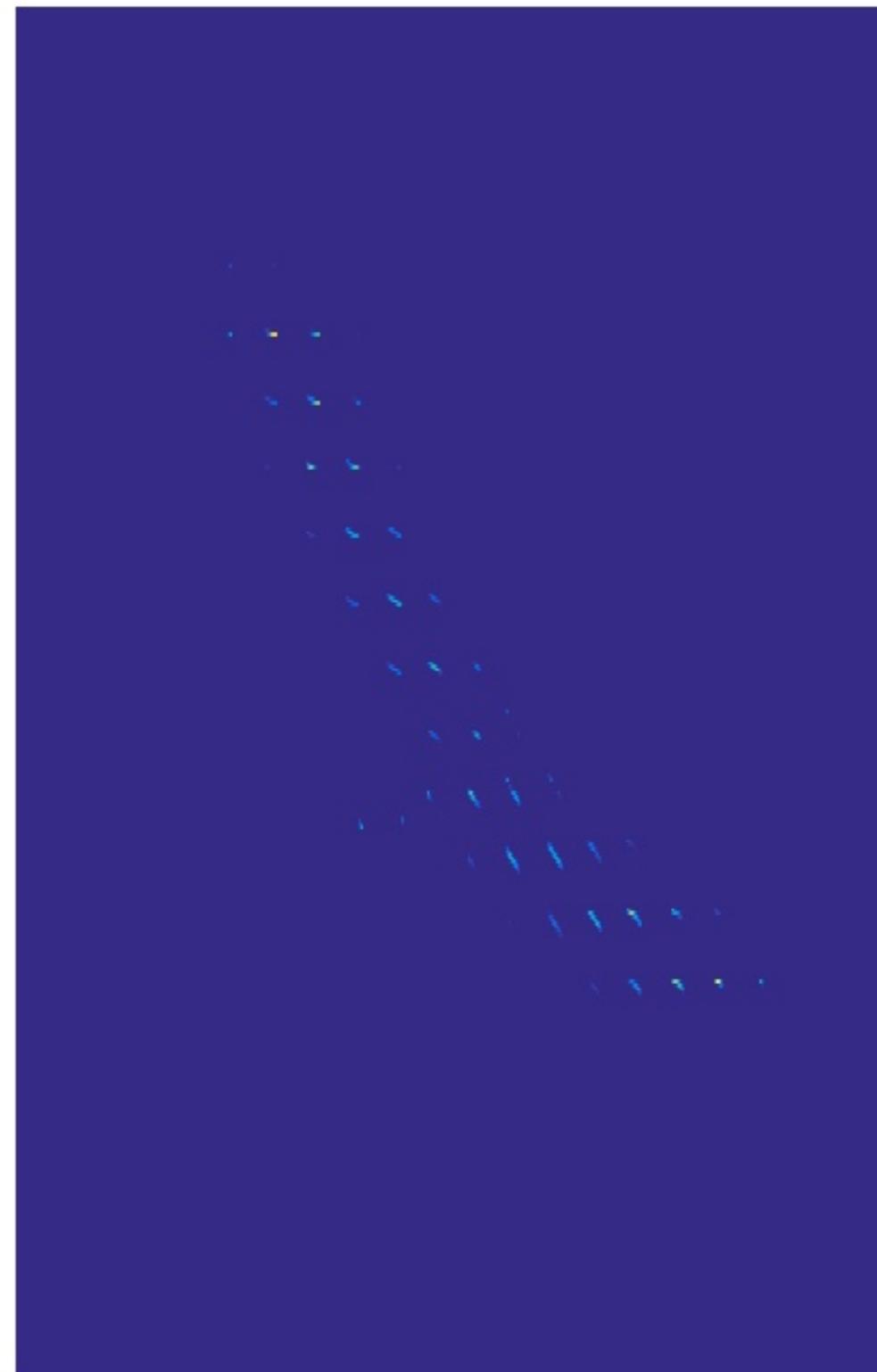


$Kv_2$



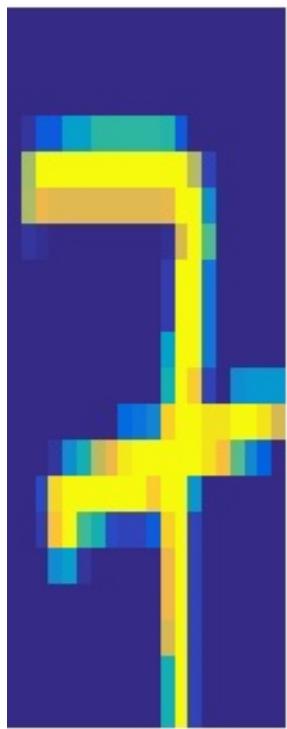
$$P_1 = D(u_1)KD(v_1)$$

$$\|P_1 - a\|_1 + \|P_1^T 1 - b\|_1 = 1.2691$$

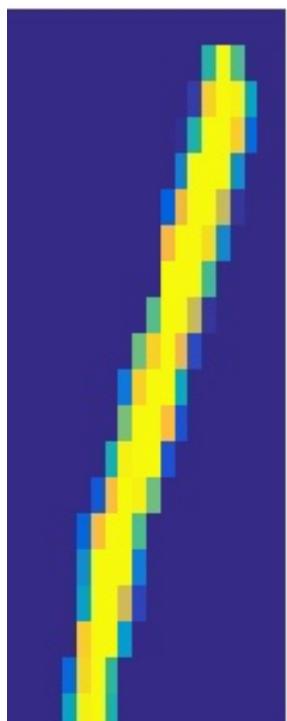


# Very Fast EMD Approx. Solver

*a*



*b*



*Ku<sub>2</sub>*

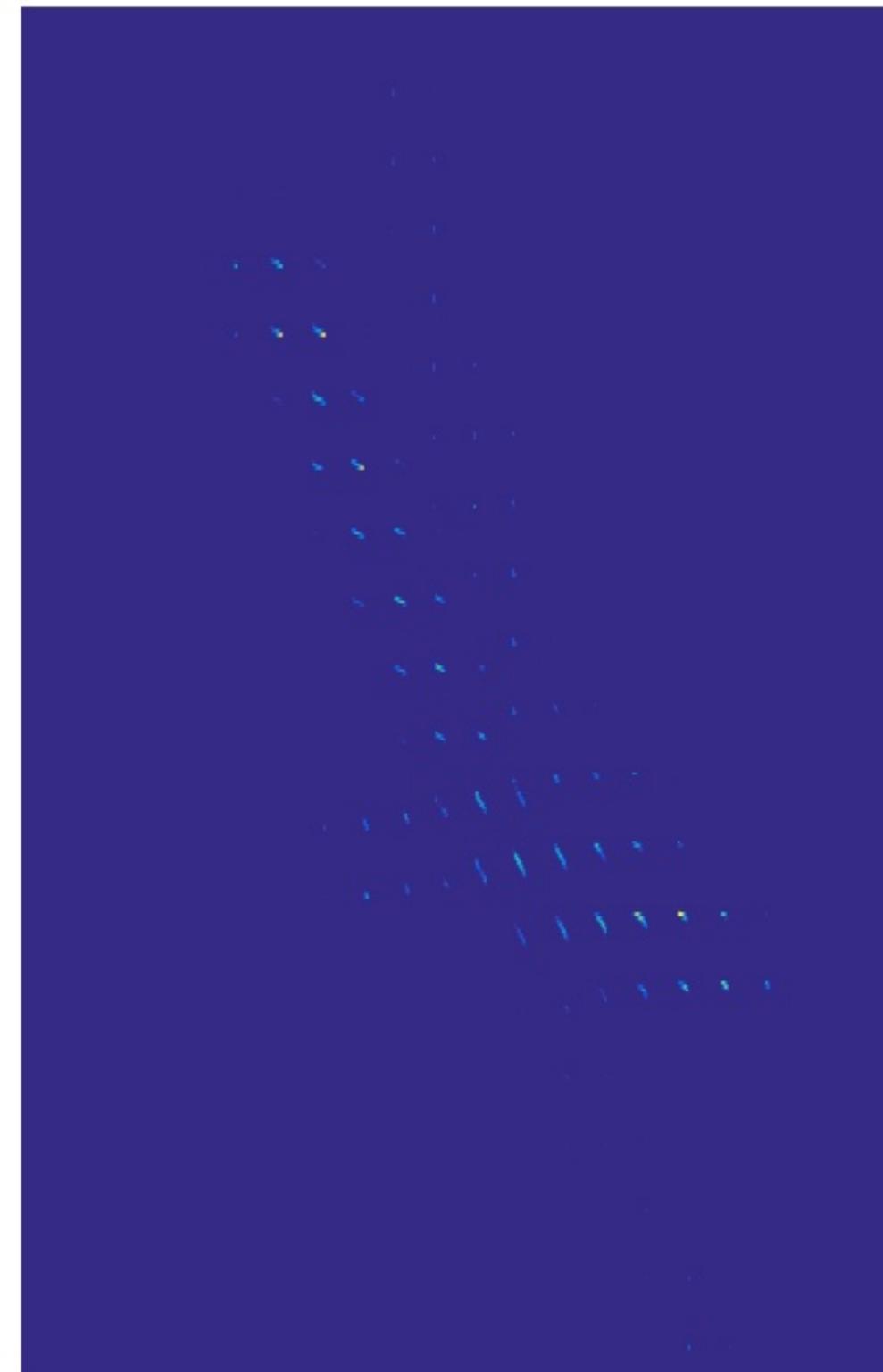


$v_3 \leftarrow b/Ku_3$



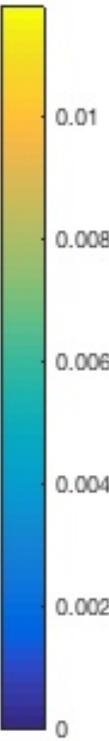
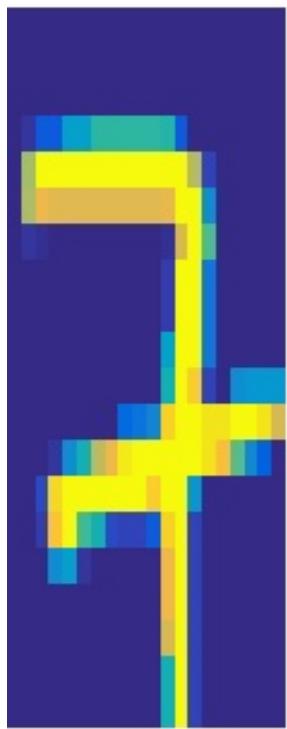
$$P_2 = D(u_2)KD(v_2)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$$

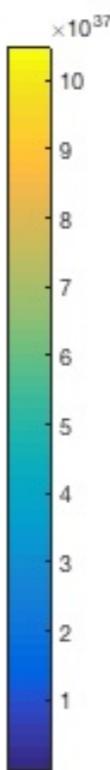


# Very Fast EMD Approx. Solver

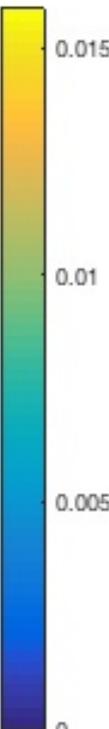
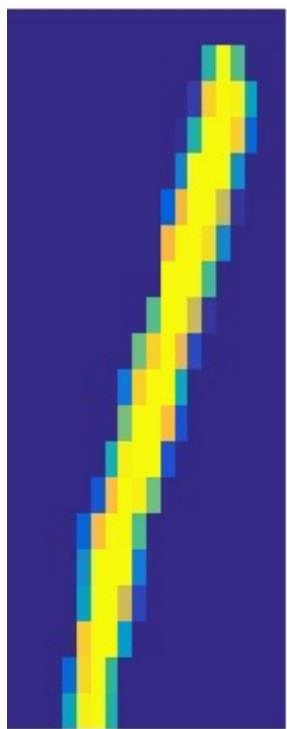
*a*



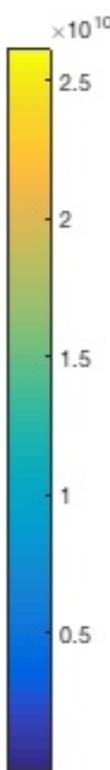
$Ku_2$



*b*

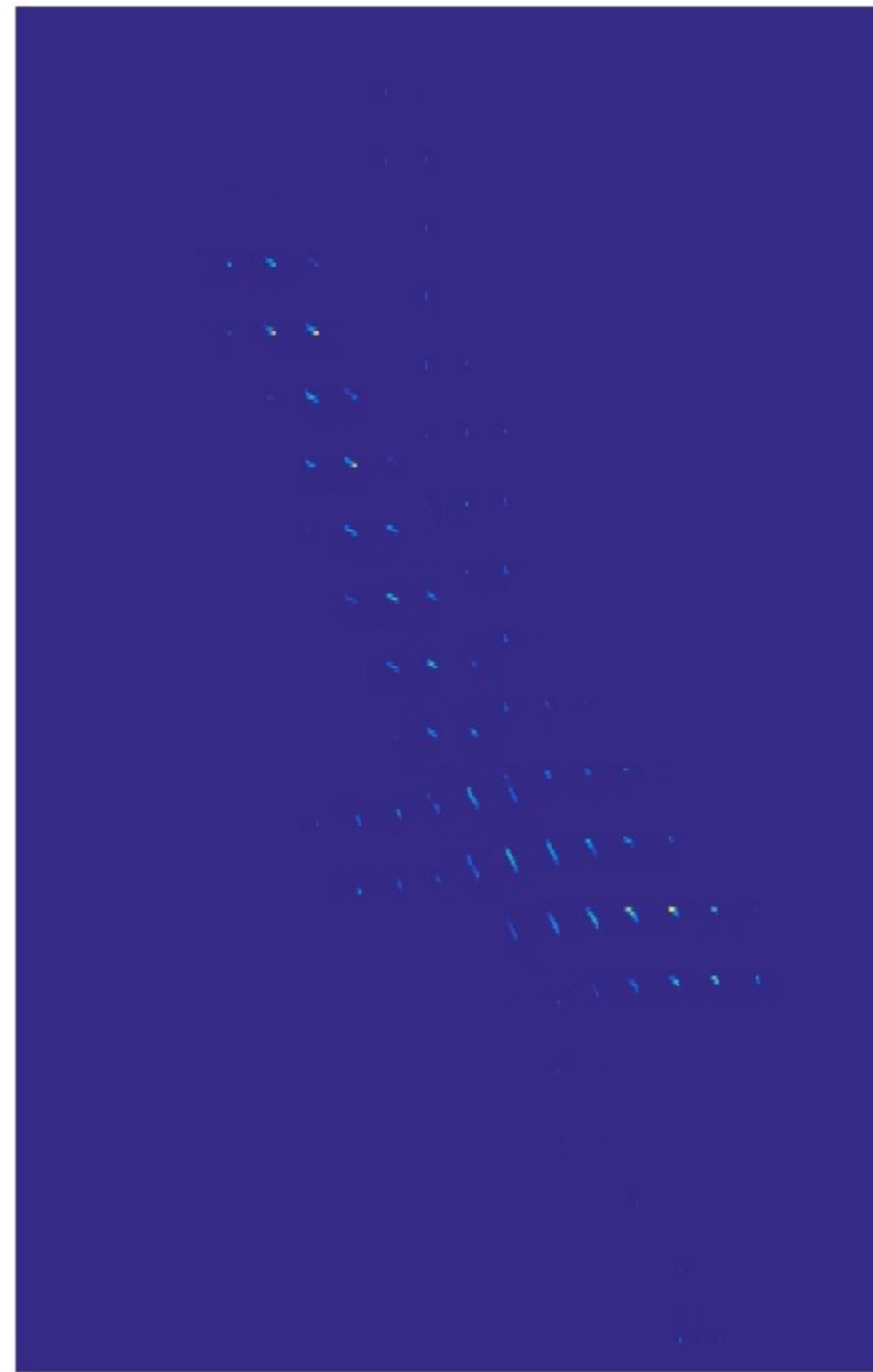


$Kv_3$



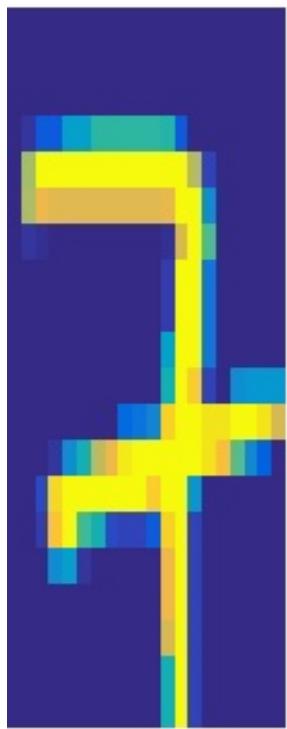
$$P_2 = D(u_2)KD(v_2)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$$

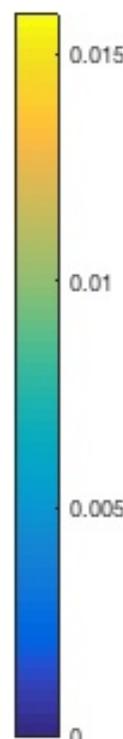


# Very Fast EMD Approx. Solver

*a*



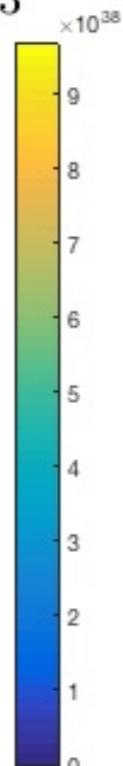
*b*



$$u_3 \leftarrow a/Kv_3$$

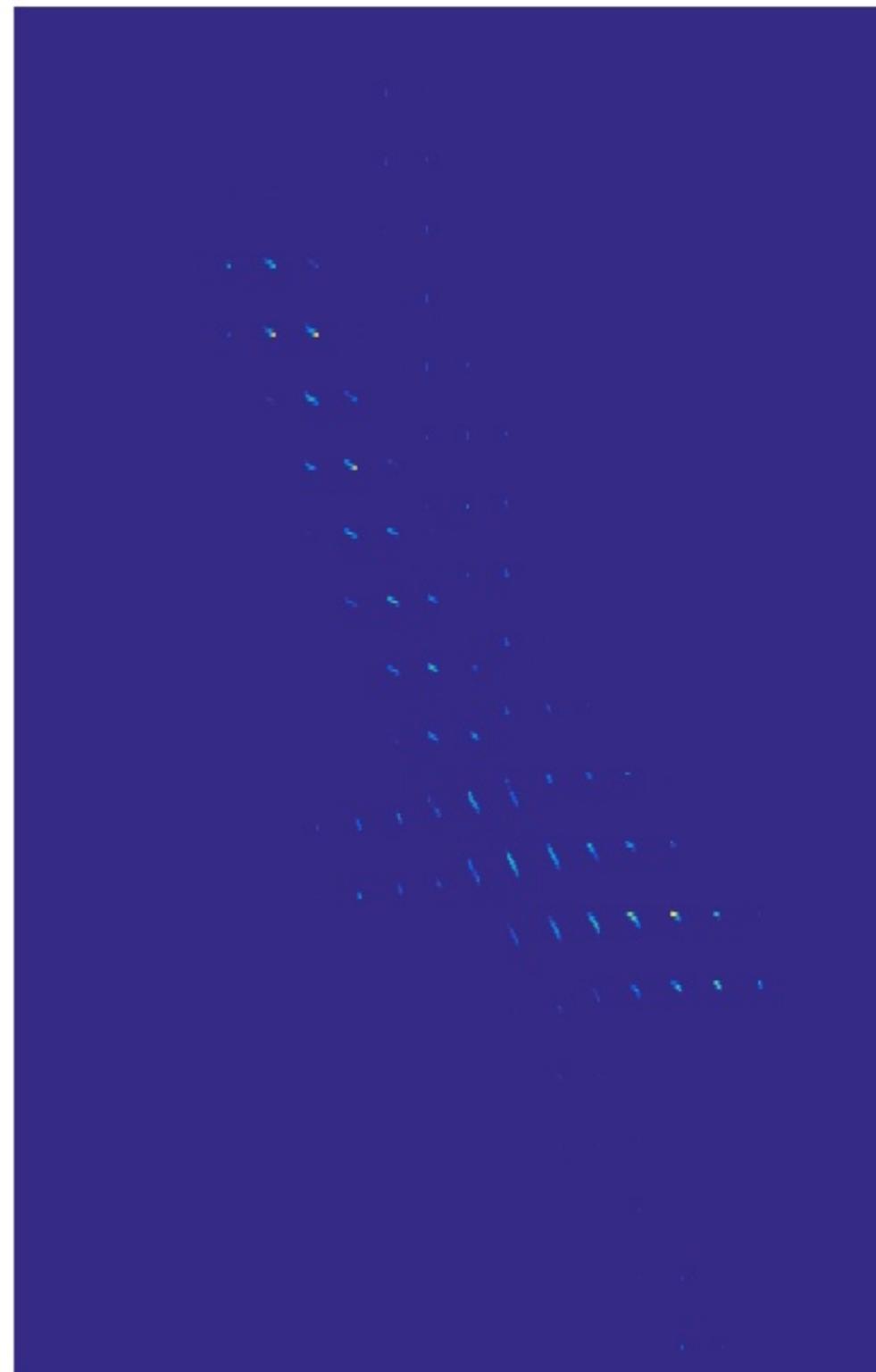


$Kv_3$



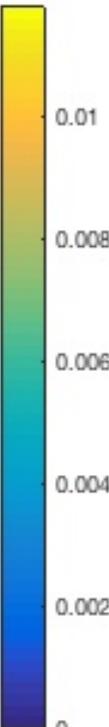
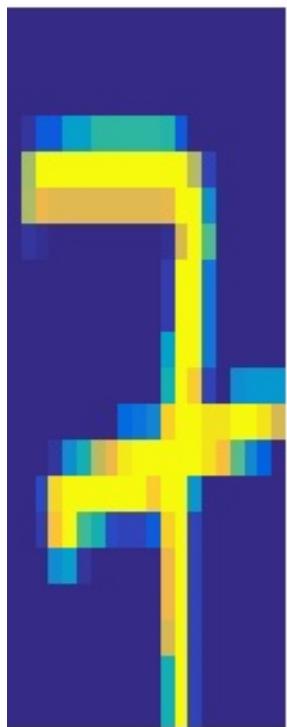
$$P_2 = D(u_2)KD(v_2)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$$

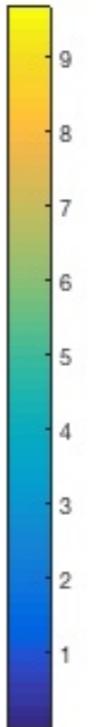


# Very Fast EMD Approx. Solver

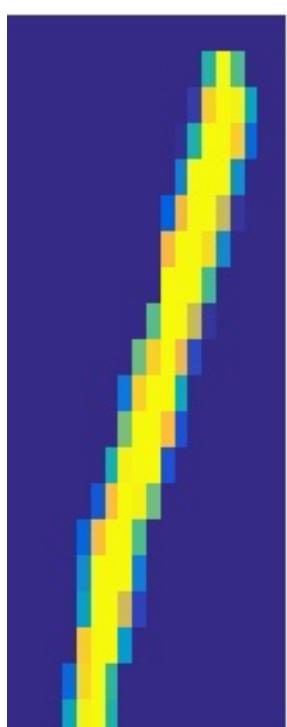
*a*



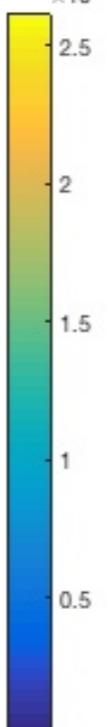
$Ku_3$



*b*

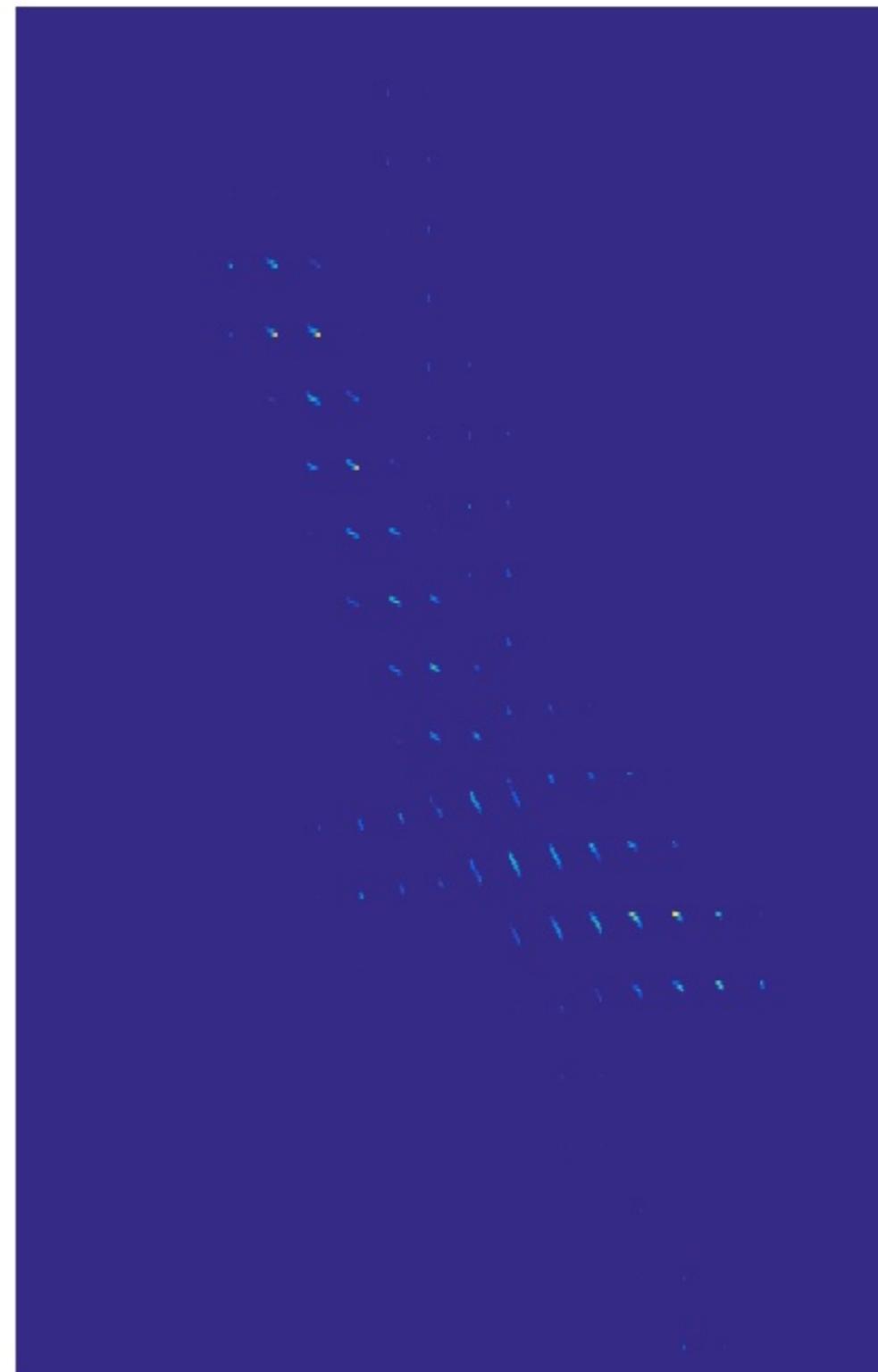


$Kv_3$



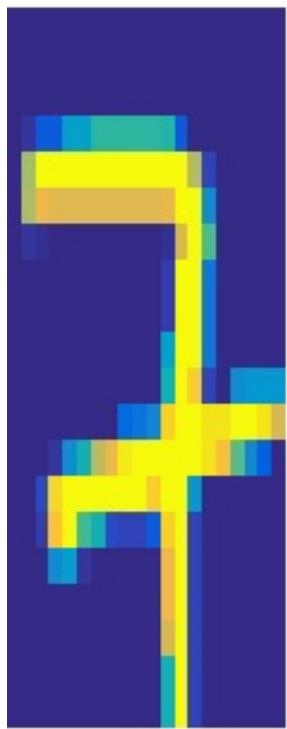
$$P_2 = D(u_2)KD(v_2)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.91067$$

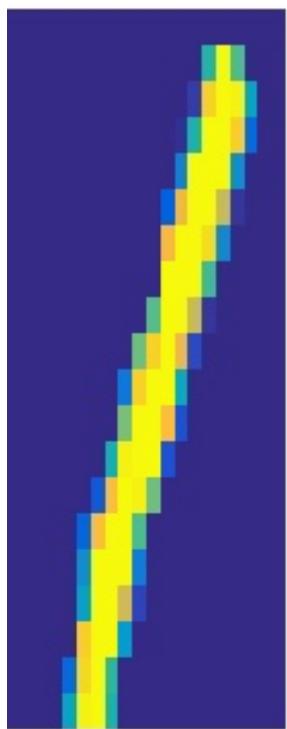


# Very Fast EMD Approx. Solver

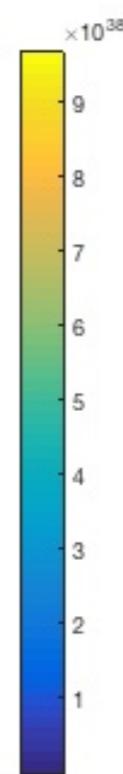
*a*



*b*

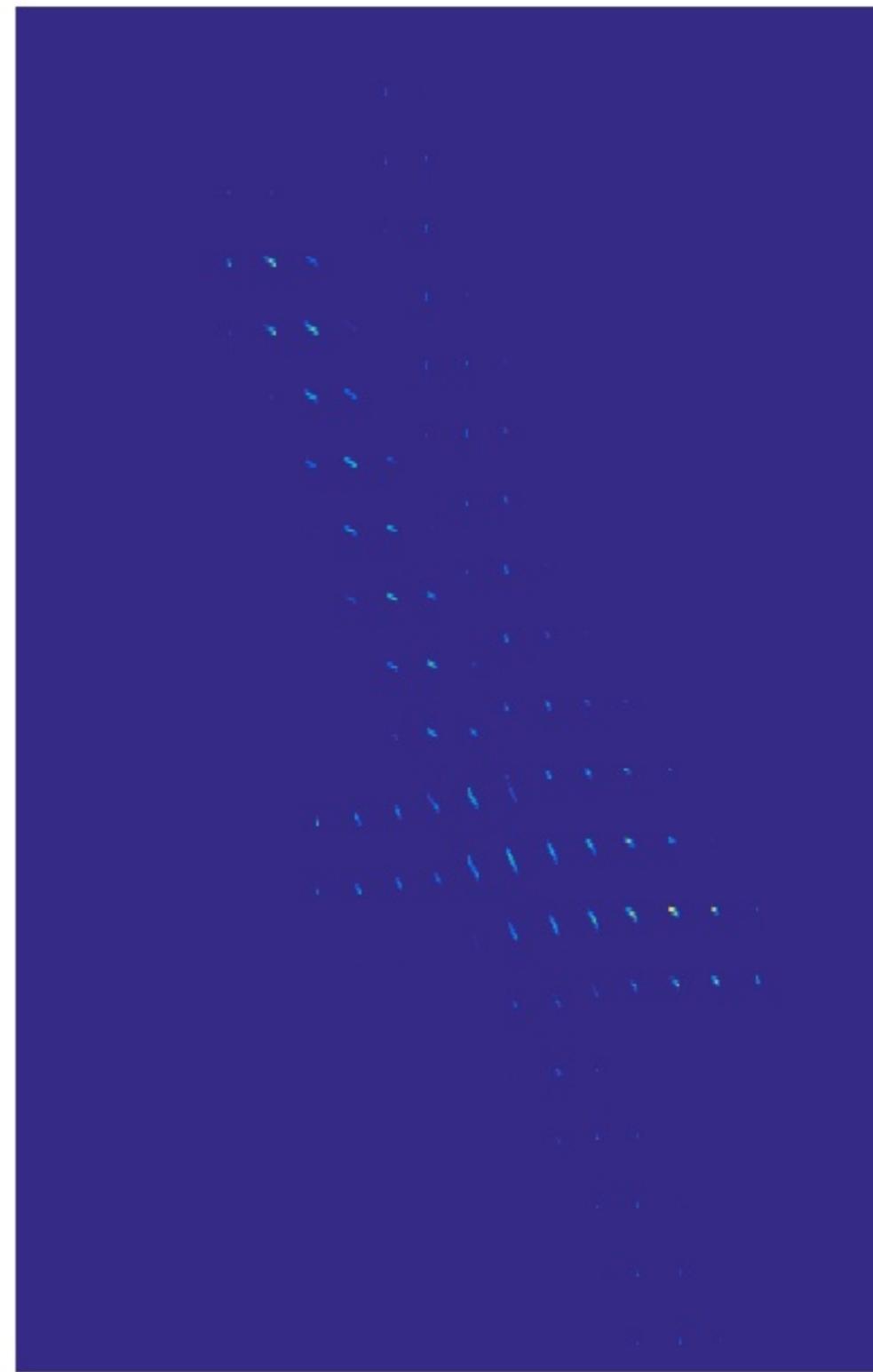


*Ku*<sub>3</sub>



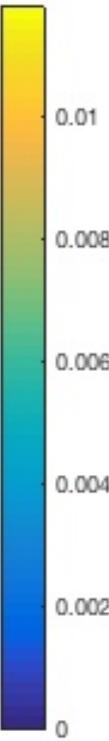
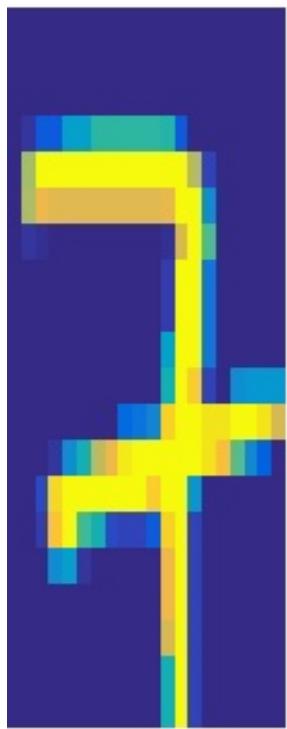
$$P_3 = D(u_3)KD(v_3)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$$

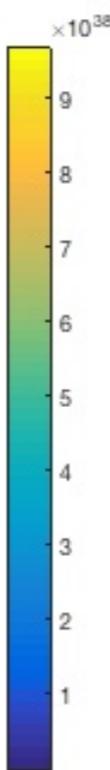


# Very Fast EMD Approx. Solver

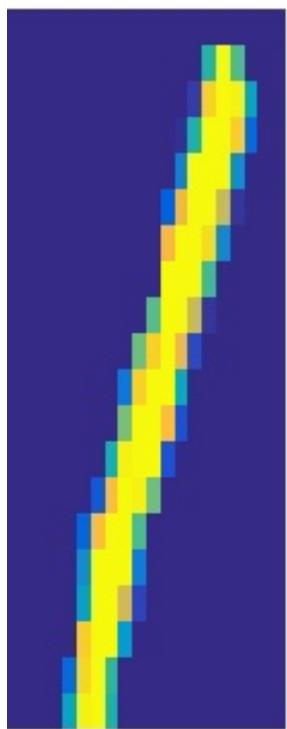
*a*



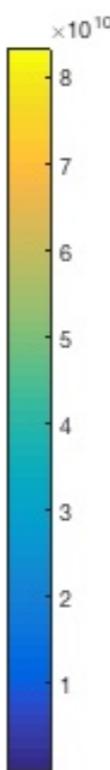
$Ku_3$



*b*

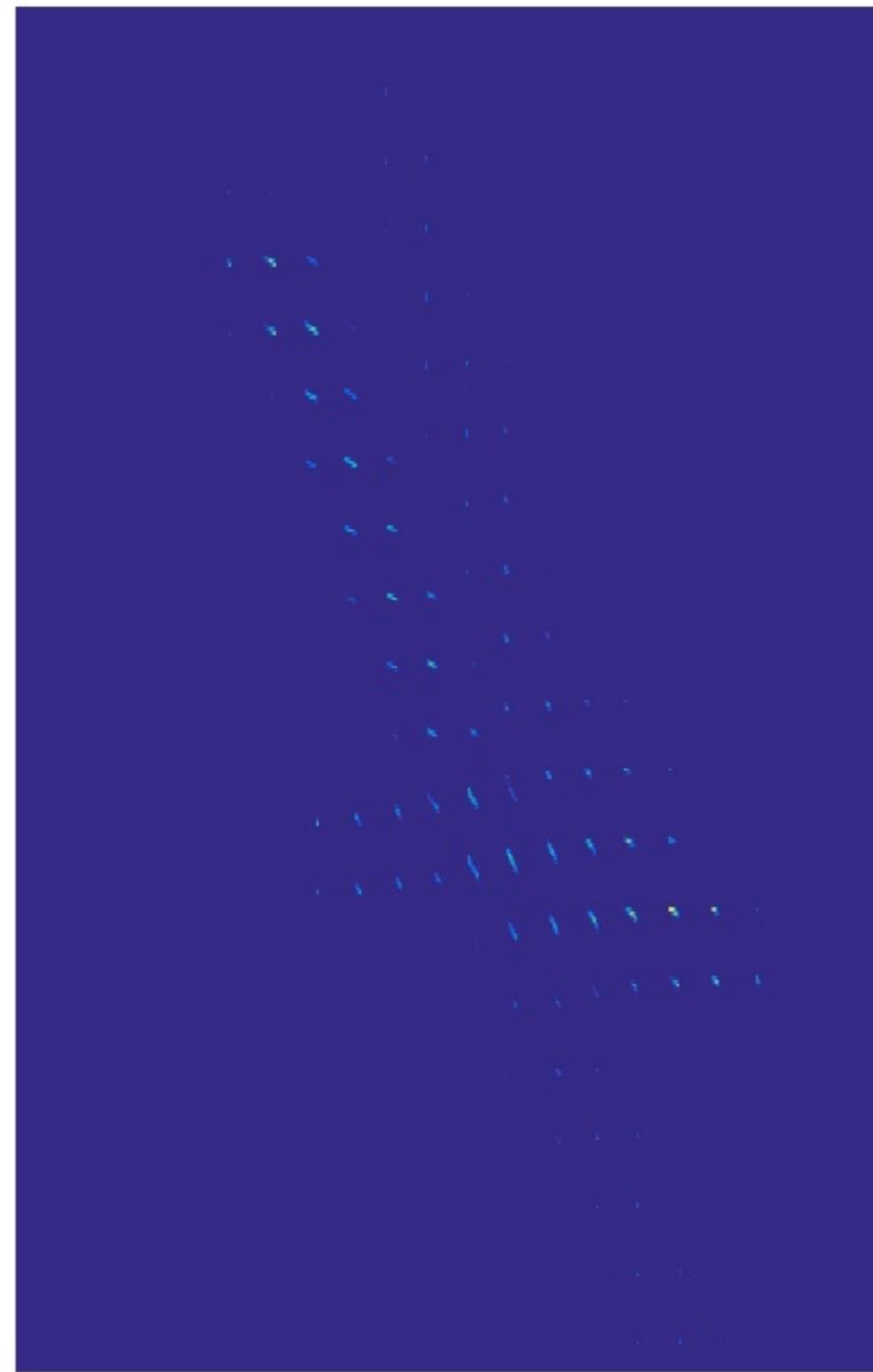


$Kv_4$



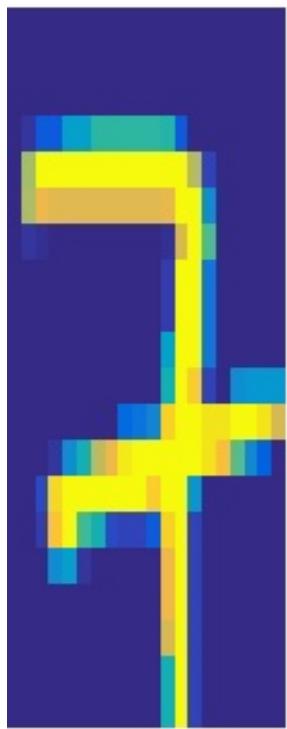
$$P_3 = D(u_3)KD(v_3)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$$

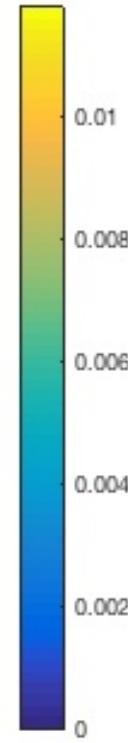


# Very Fast EMD Approx. Solver

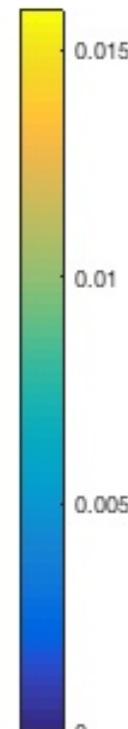
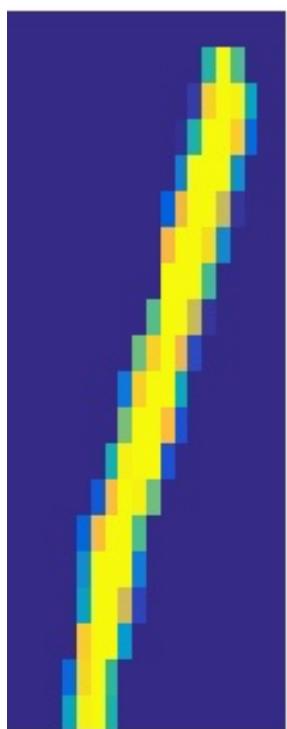
*a*



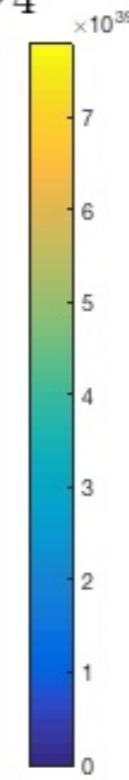
$$u_4 \leftarrow a/Kv_4$$



*b*

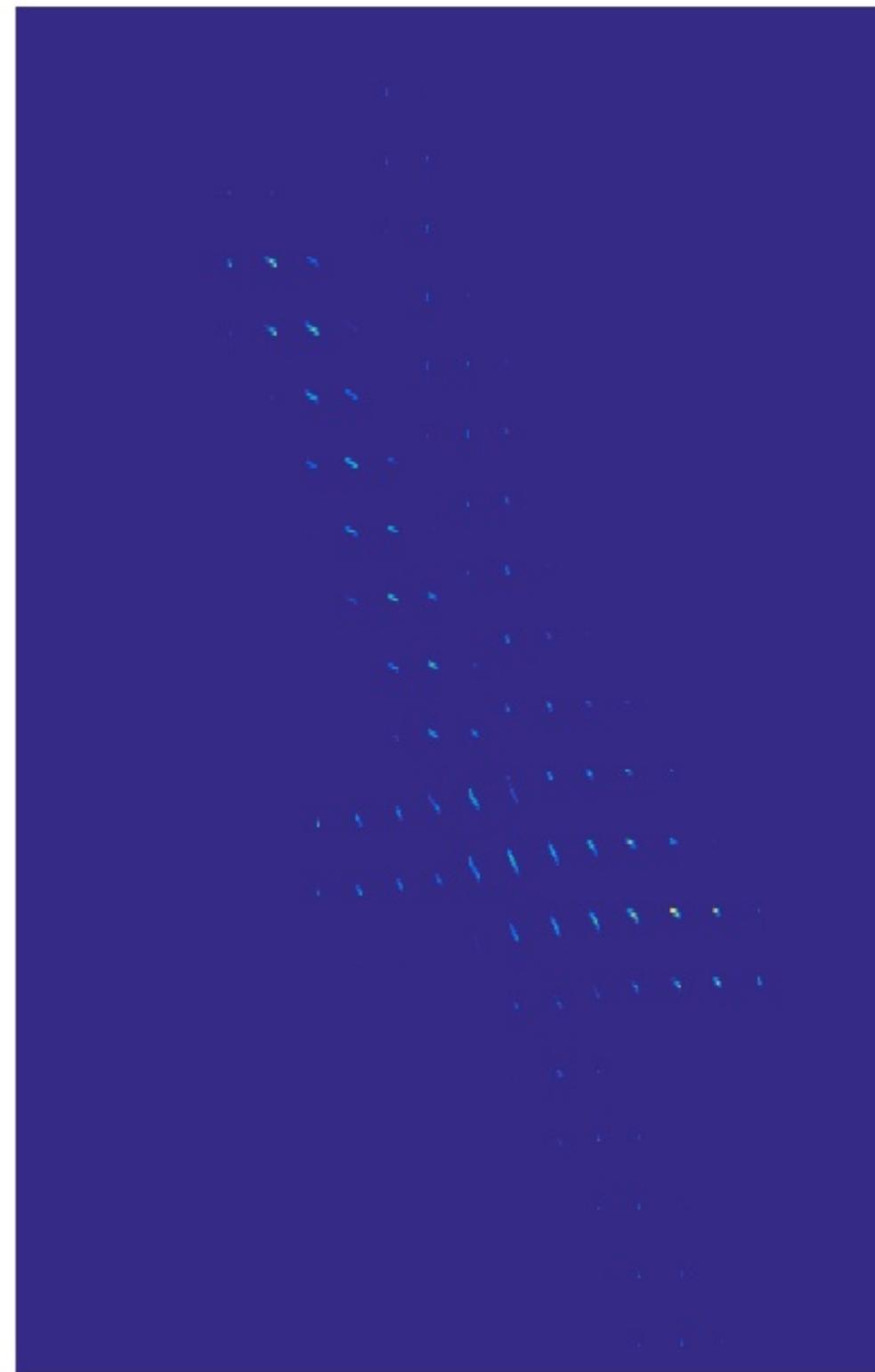


$Kv_4$



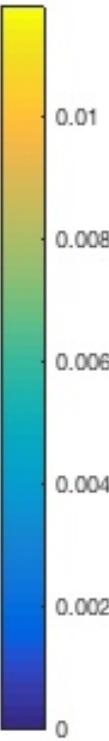
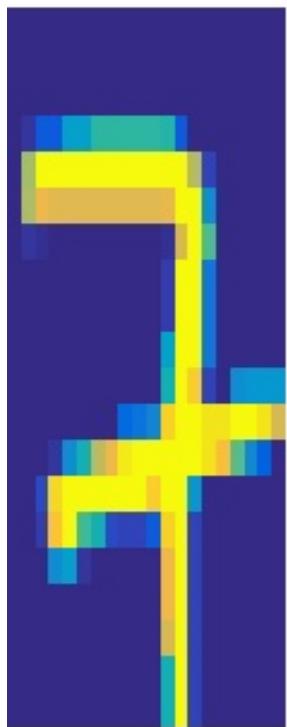
$$P_3 = D(u_3)KD(v_3)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$$

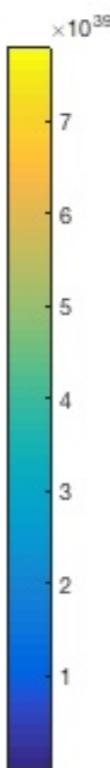


# Very Fast EMD Approx. Solver

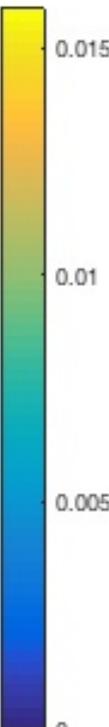
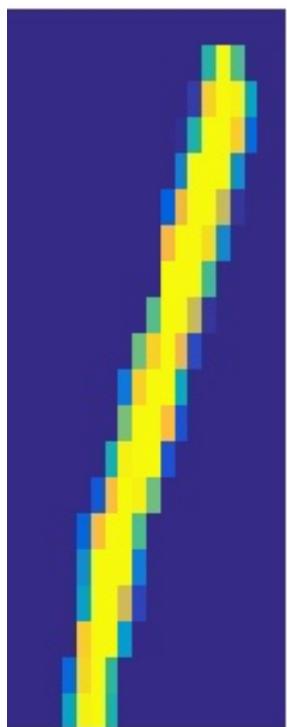
*a*



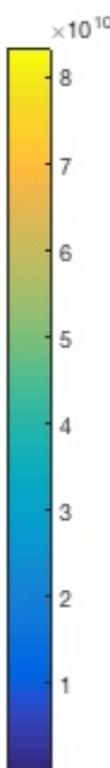
$Ku_4$



*b*

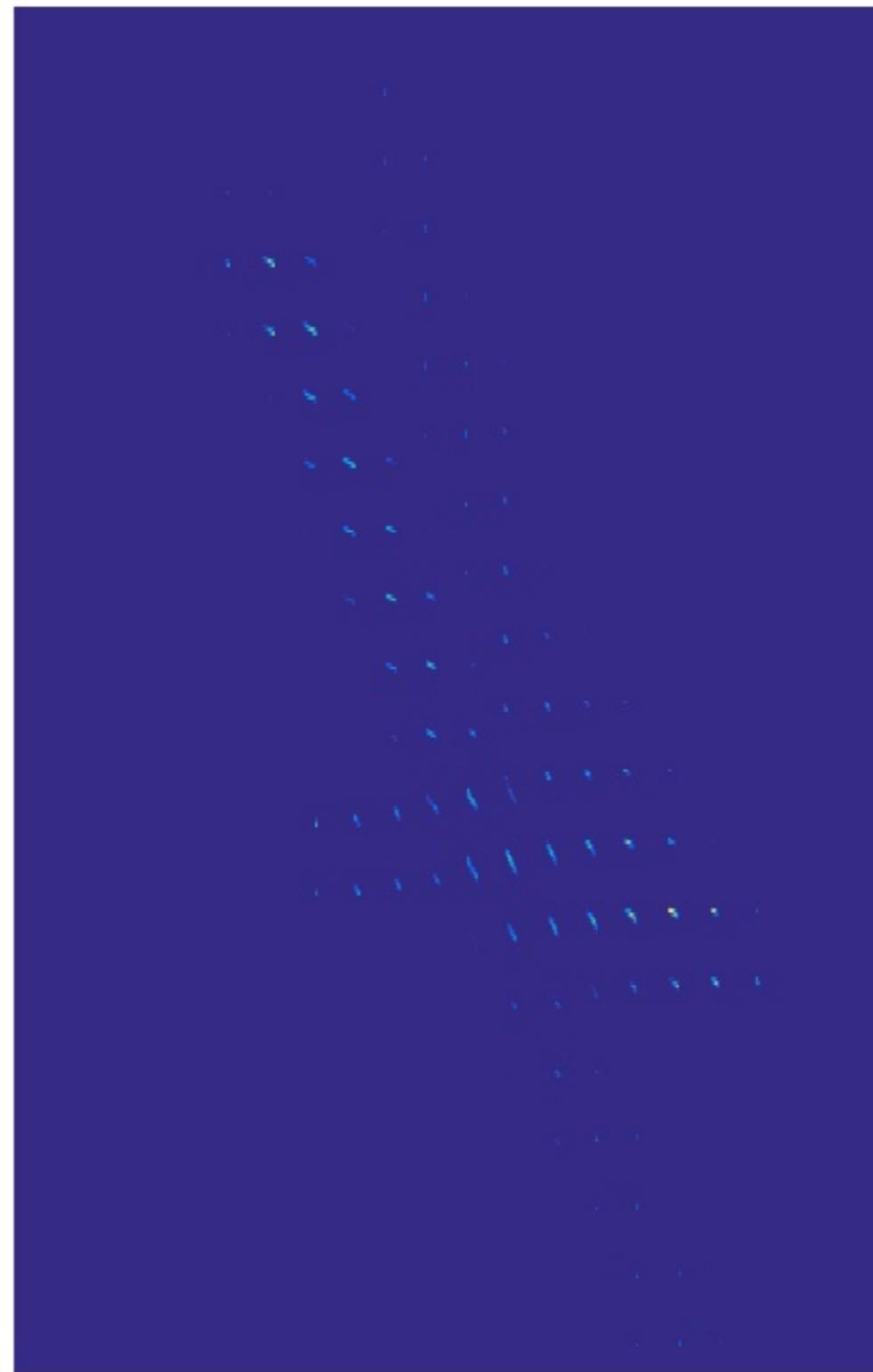


$Kv_4$



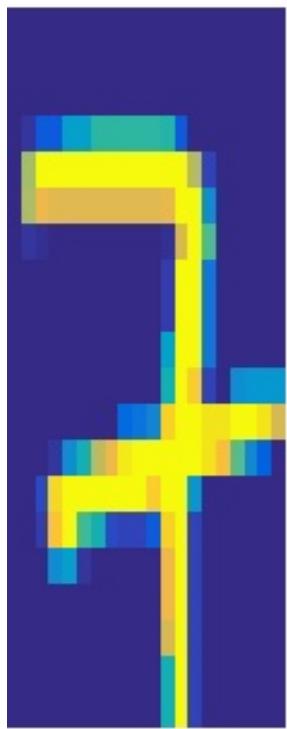
$$P_3 = D(u_3)KD(v_3)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.70387$$

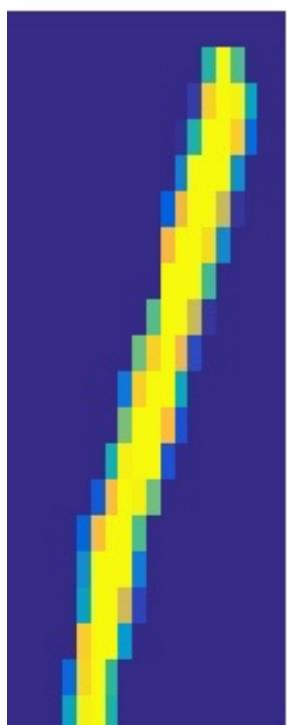


# Very Fast EMD Approx. Solver

*a*



*b*



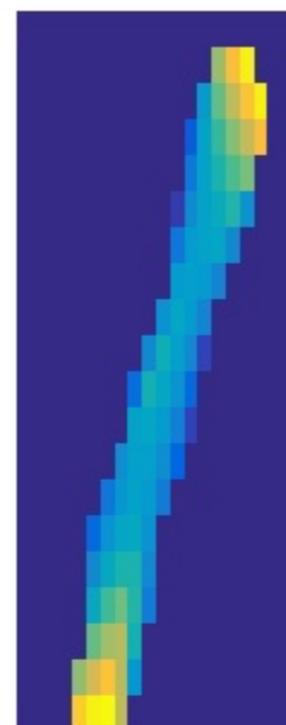
*Ku*<sub>4</sub>



$v_5 \leftarrow b/Ku_5$

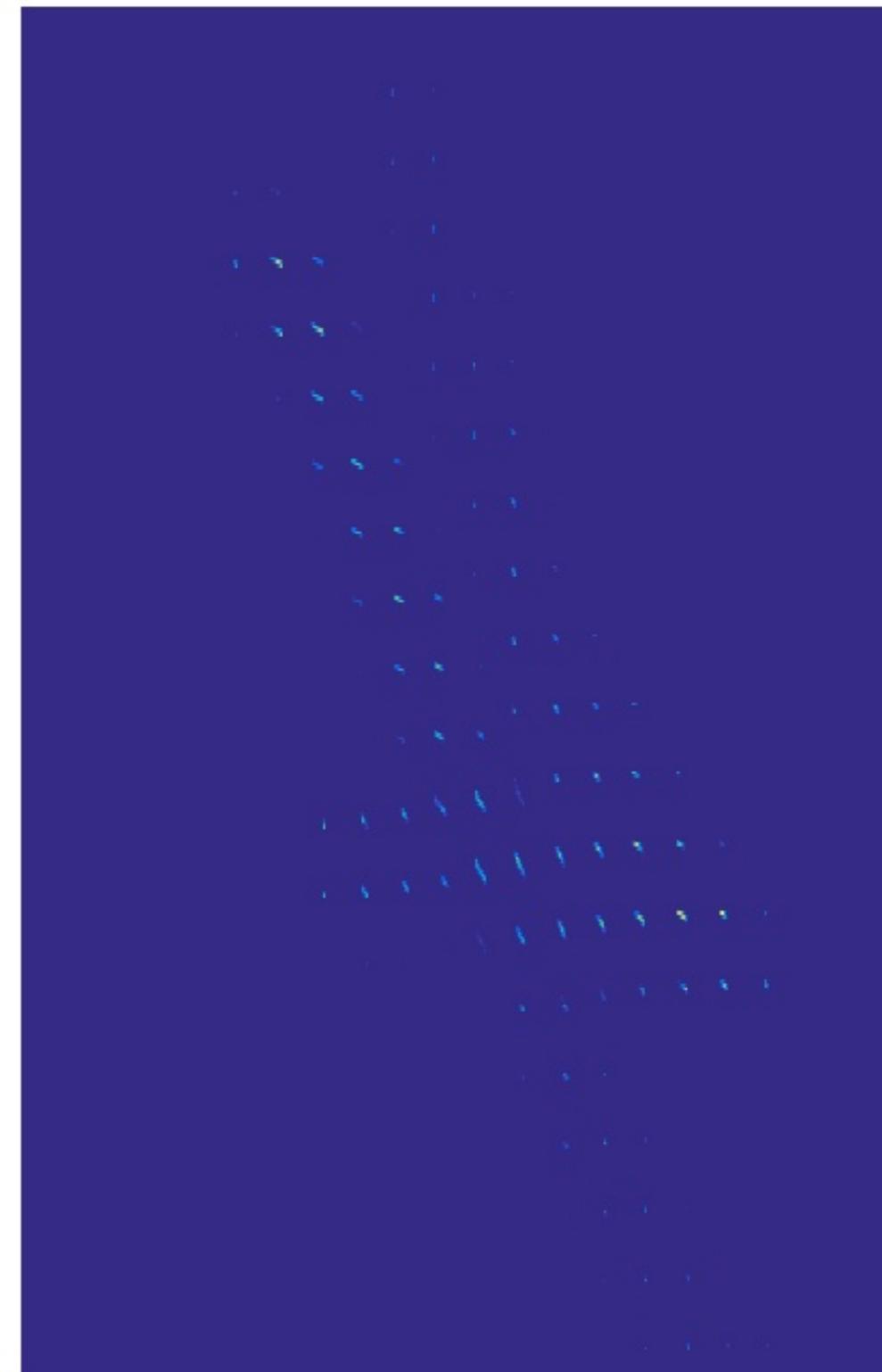


$\log(v_5)$



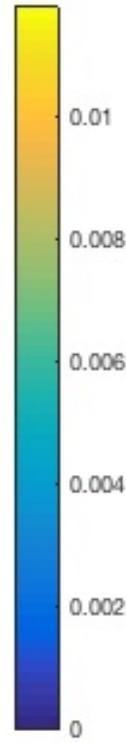
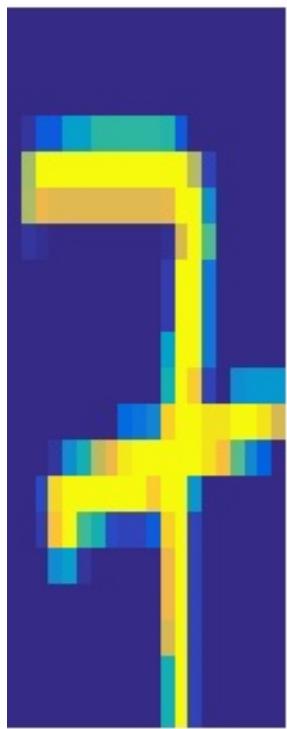
$$P_4 = D(u_4)KD(v_4)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$

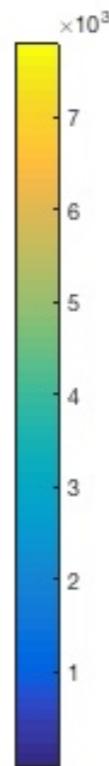


# Very Fast EMD Approx. Solver

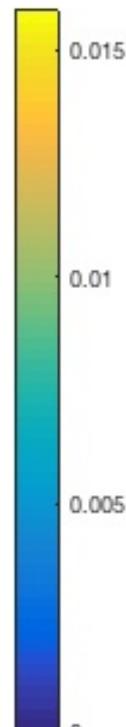
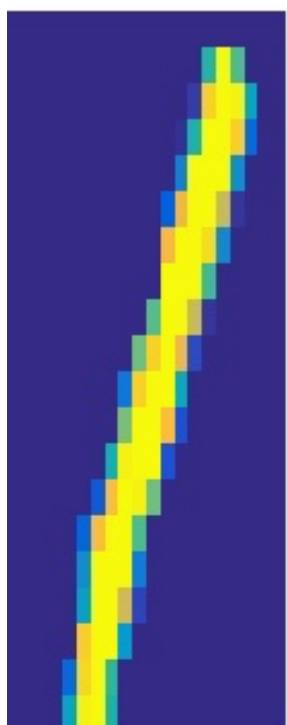
*a*



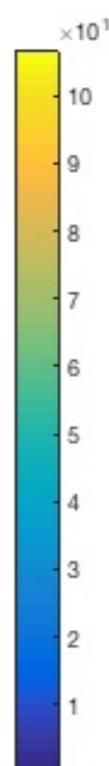
$Ku_4$



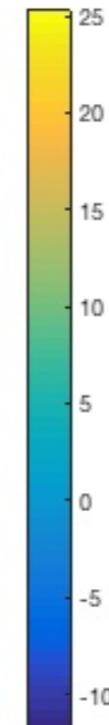
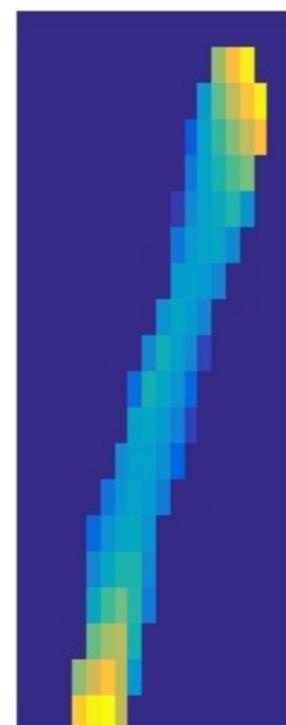
*b*



$Kv_5$

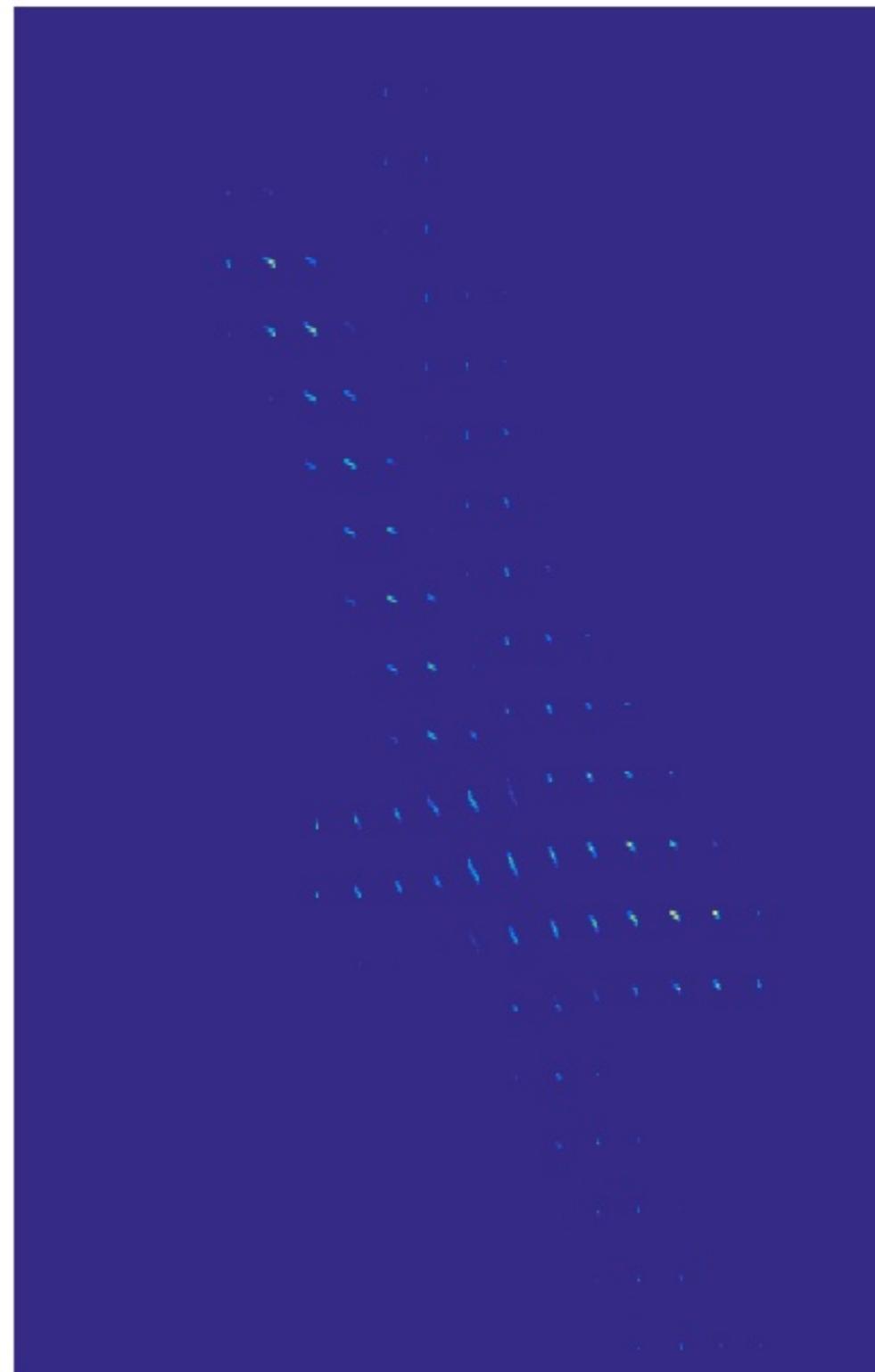


$\log(v_5)$



$$P_4 = D(u_4)KD(v_4)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$



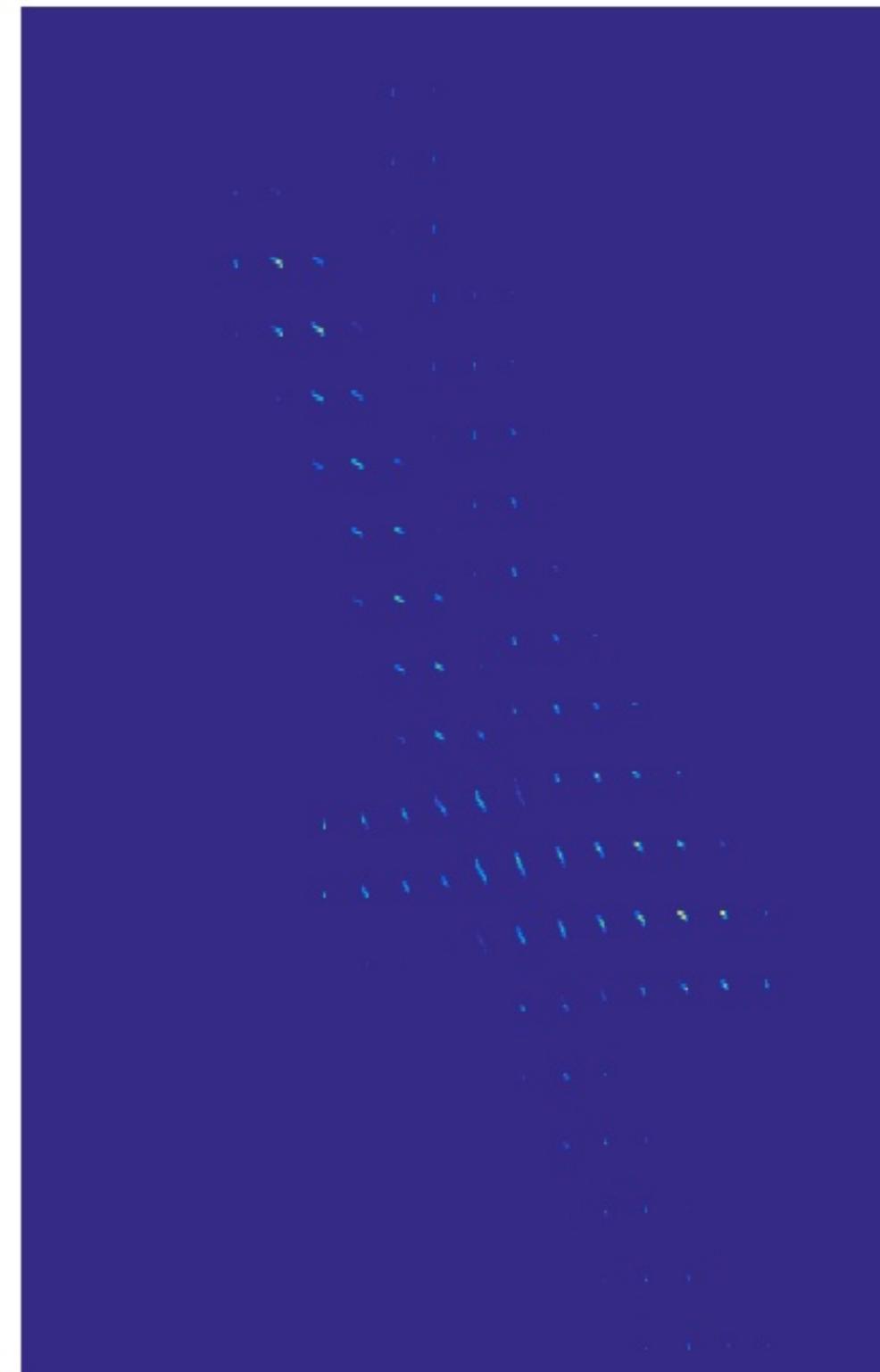
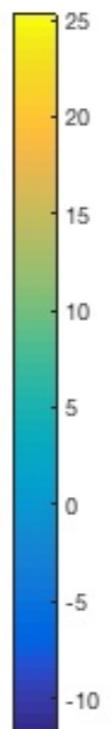
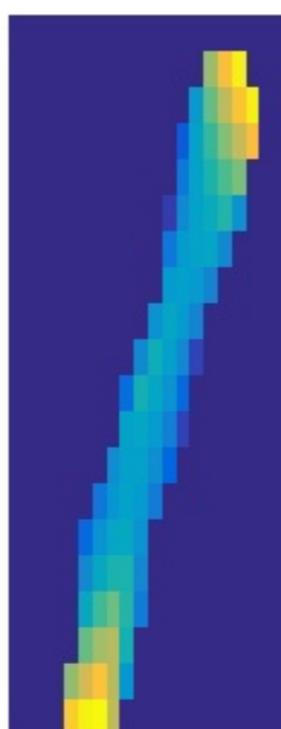
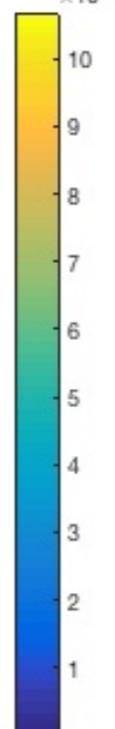
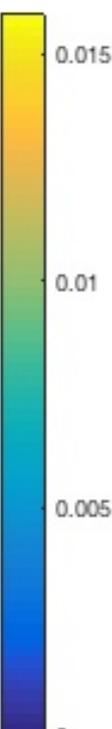
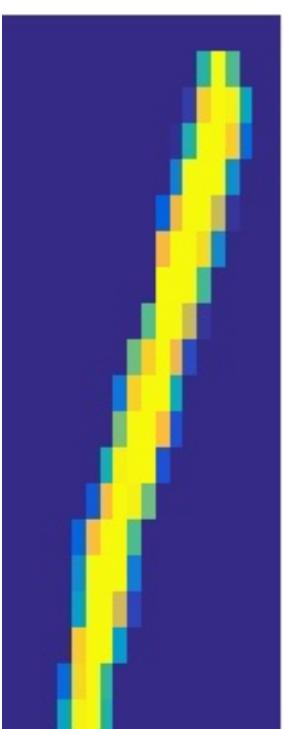
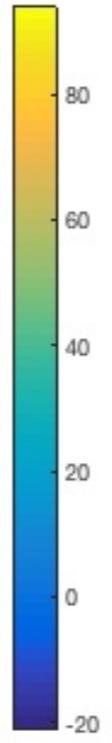
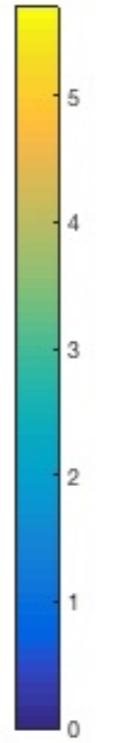
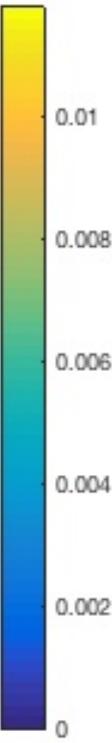
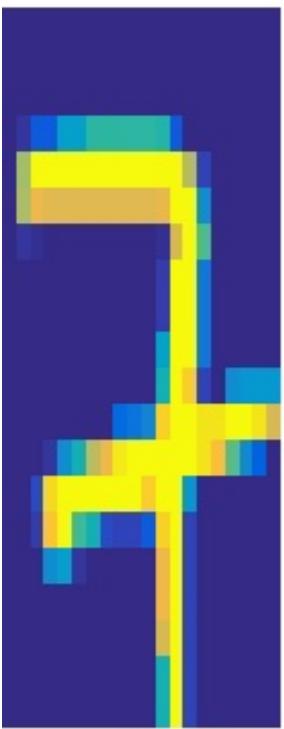
# Very Fast EMD Approx. Solver

*a*

$$u_5 \leftarrow a/Kv_5$$

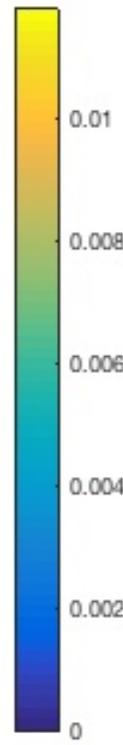
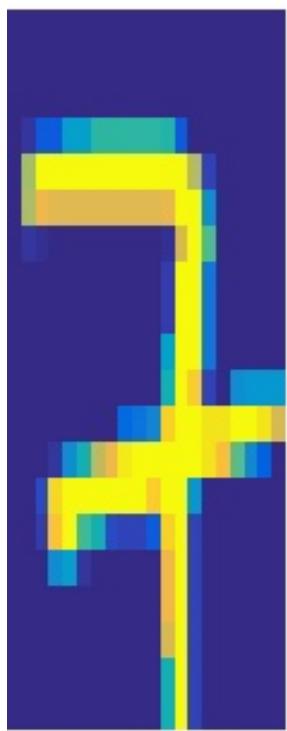
$$\log(u_5)$$

$$P_4 = D(u_4)KD(v_4)$$
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$

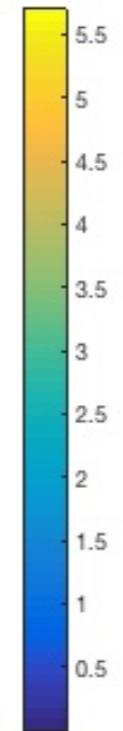


# Very Fast EMD Approx. Solver

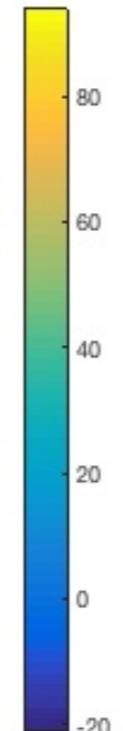
*a*



$Ku_5$



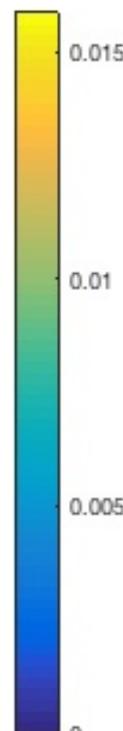
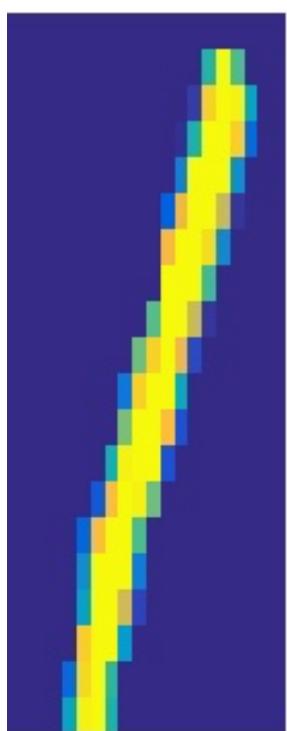
$\log(u_5)$



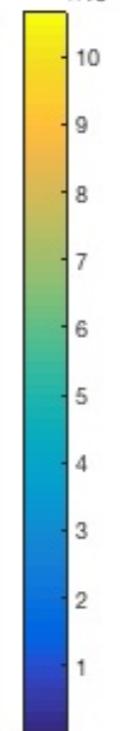
$$P_4 = D(u_4)KD(v_4)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.58736$$

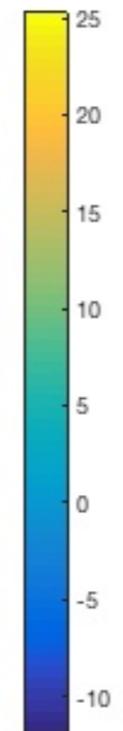
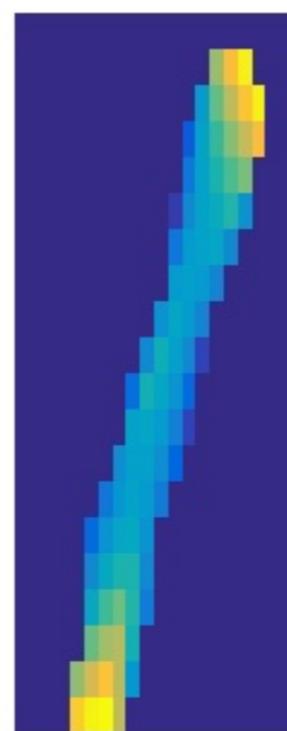
*b*



$Kv_5$

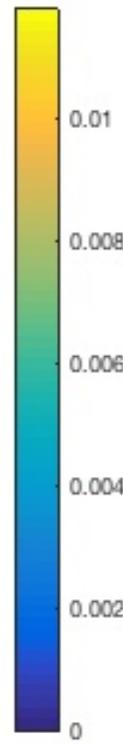
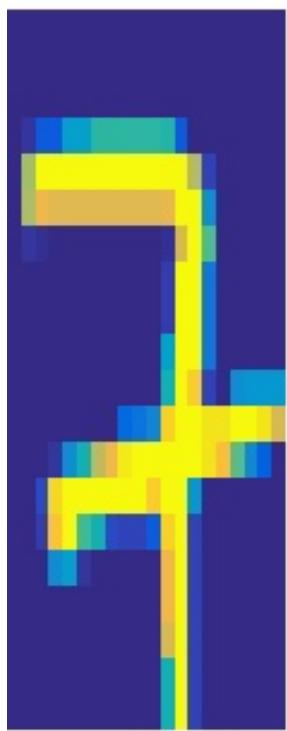


$\log(v_5)$

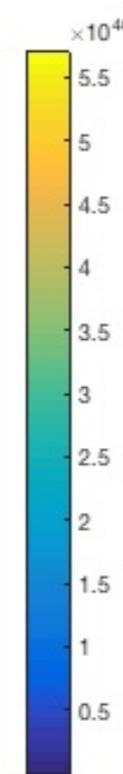


# Very Fast EMD Approx. Solver

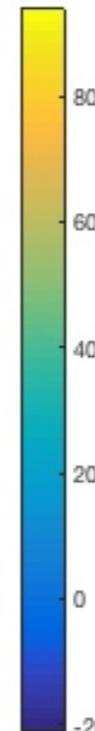
*a*



*Ku<sub>5</sub>*



$\log(u_5)$

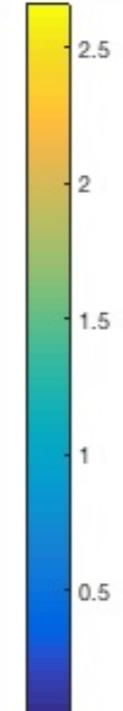
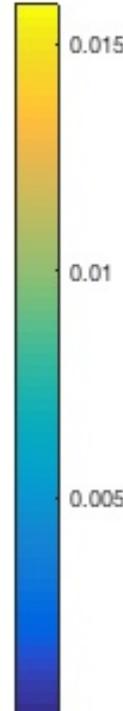
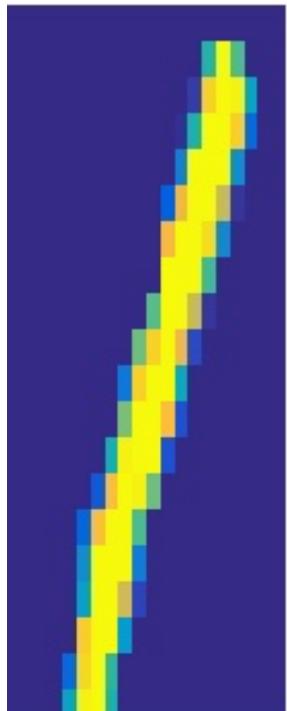


$$P_5 = D(u_5)KD(v_5)$$

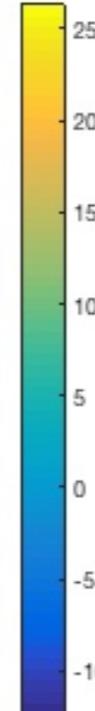
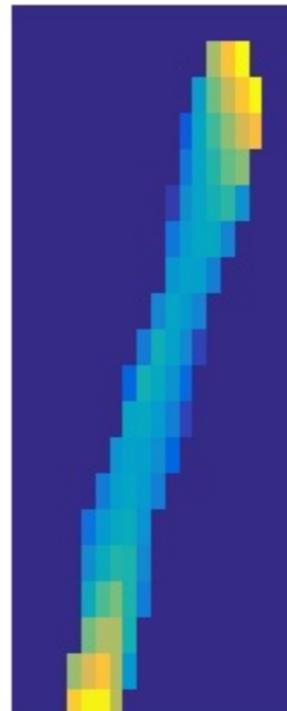
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$$

*b*

$v_6 \leftarrow b/Ku_6$

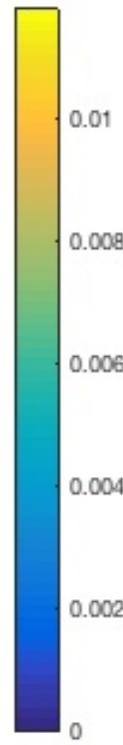
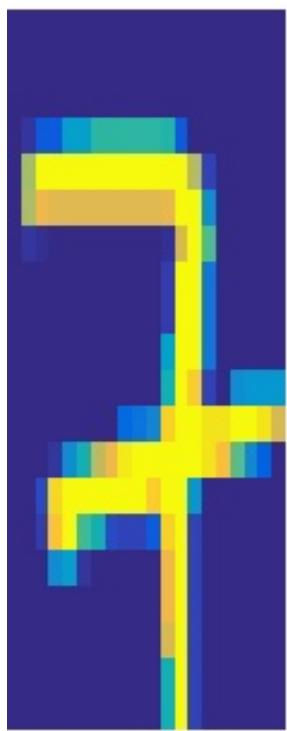


$\log(v_6)$

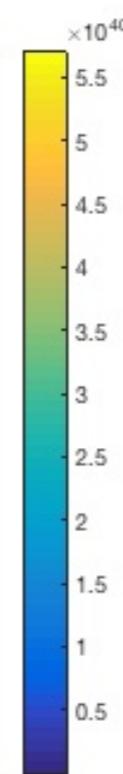


# Very Fast EMD Approx. Solver

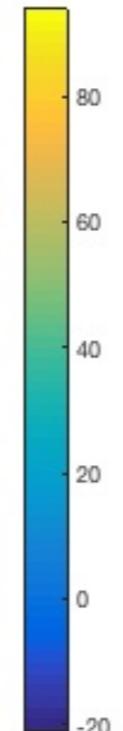
*a*



$Ku_5$



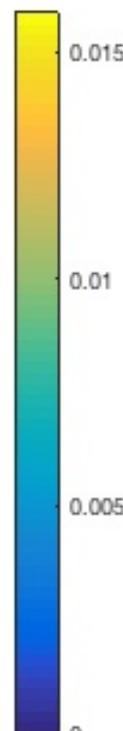
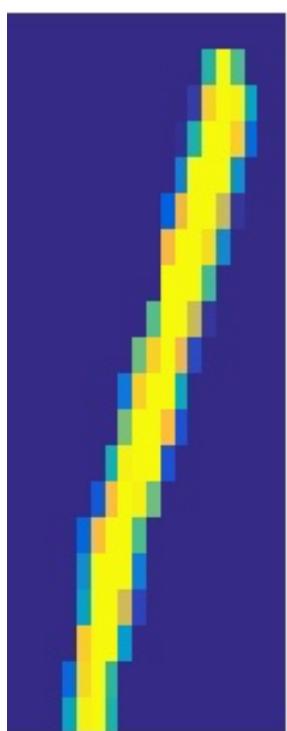
$\log(u_5)$



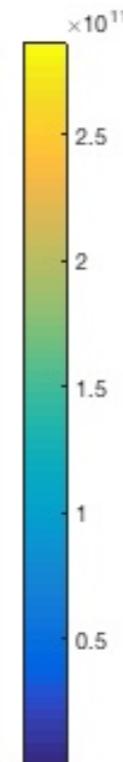
$$P_5 = D(u_5)KD(v_5)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$$

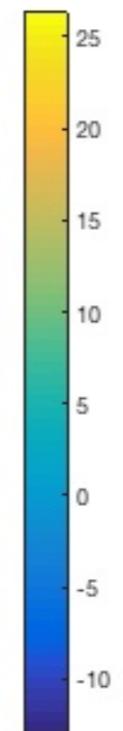
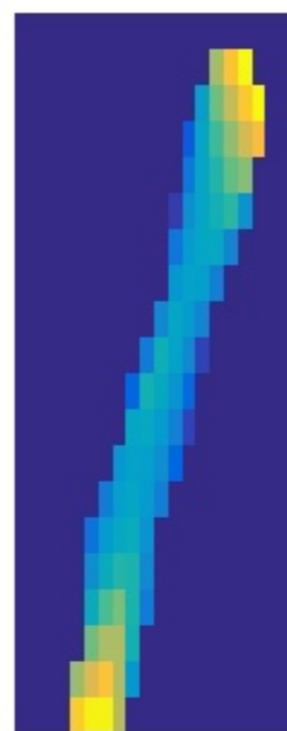
*b*



$Kv_6$

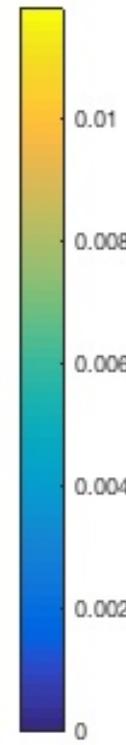
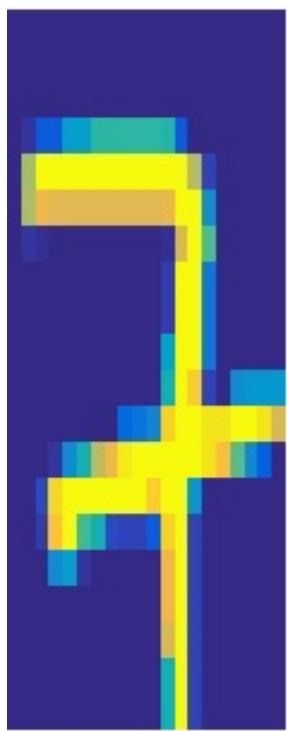


$\log(v_6)$



# Very Fast EMD Approx. Solver

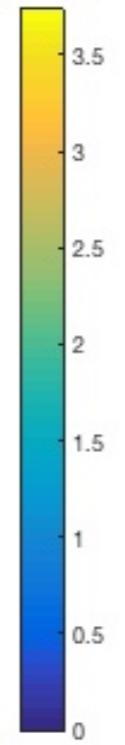
*a*



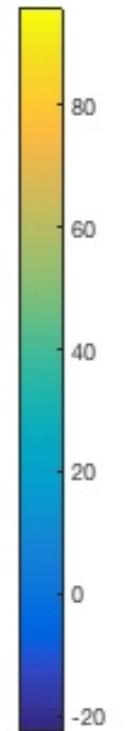
$$u_6 \leftarrow a/Kv_6$$



$\times 10^{41}$



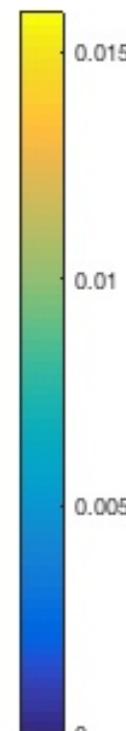
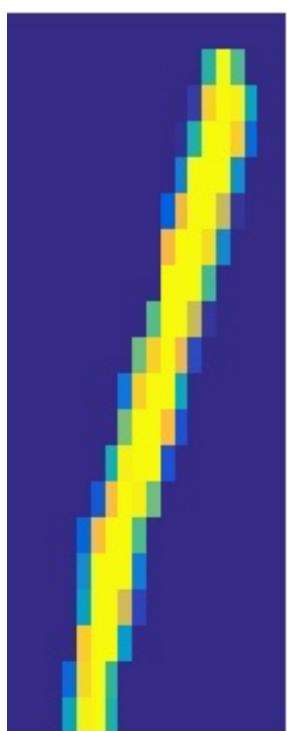
$$\log(u_6)$$



$$P_5 = D(u_5)KD(v_5)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$$

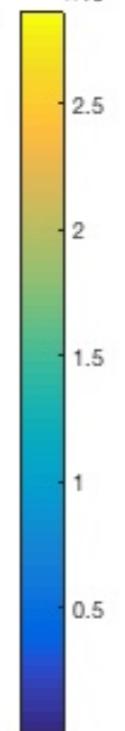
*b*



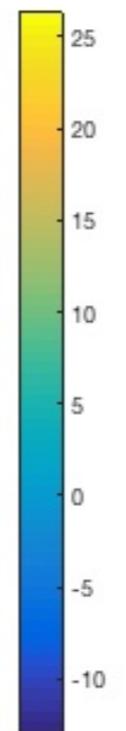
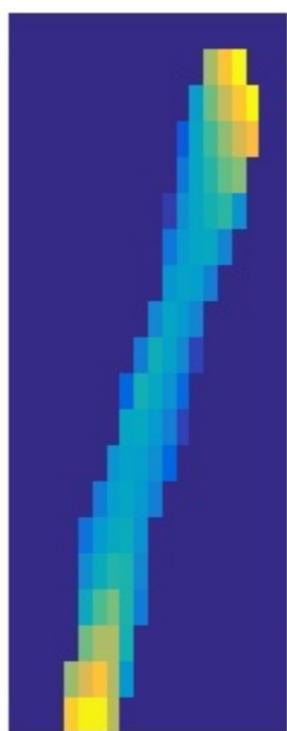
$$Kv_6$$



$\times 10^{11}$

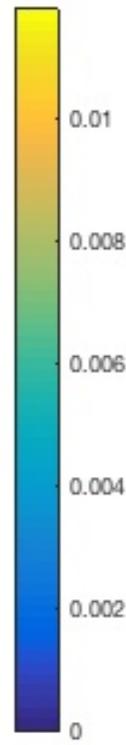
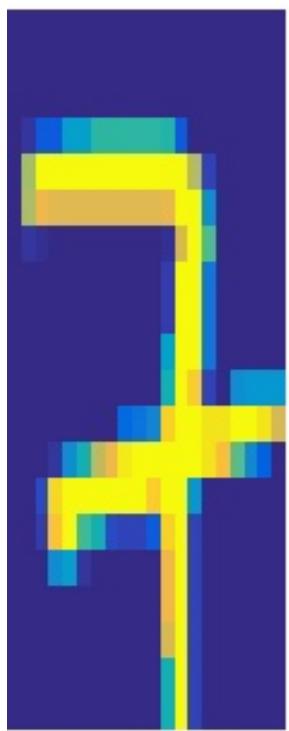


$$\log(v_6)$$

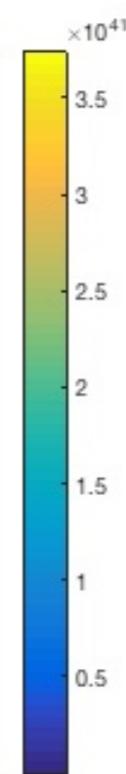


# Very Fast EMD Approx. Solver

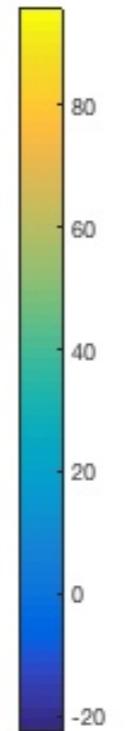
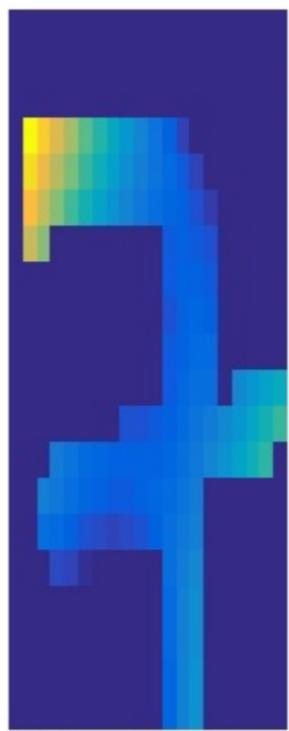
*a*



$Ku_6$



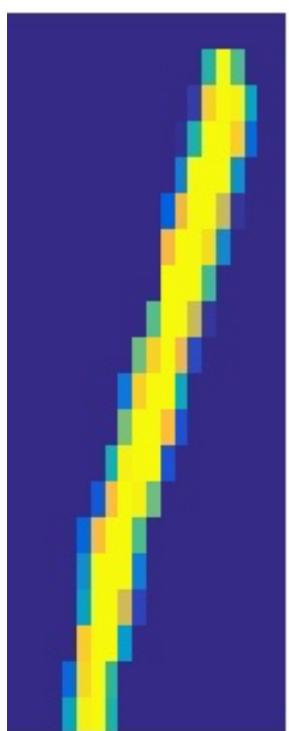
$\log(u_6)$



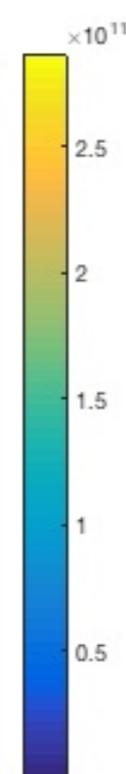
$$P_5 = D(u_5)KD(v_5)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.50974$$

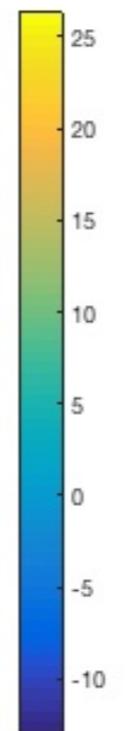
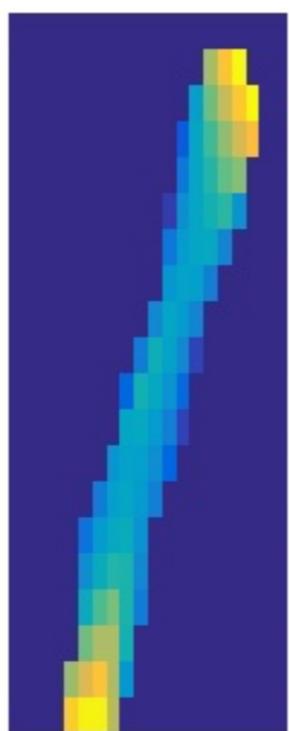
*b*



$Kv_6$

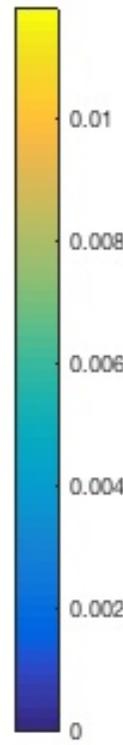
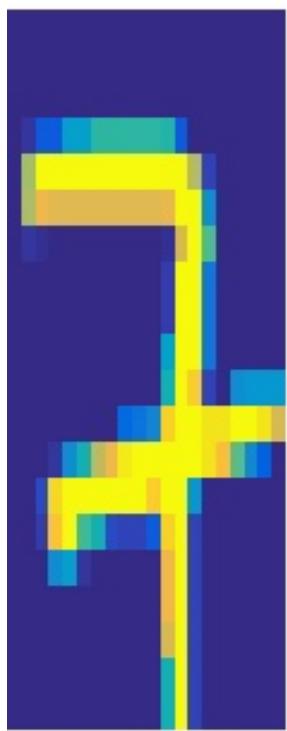


$\log(v_6)$

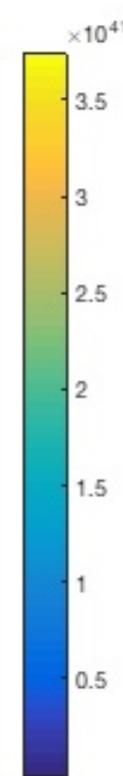


# Very Fast EMD Approx. Solver

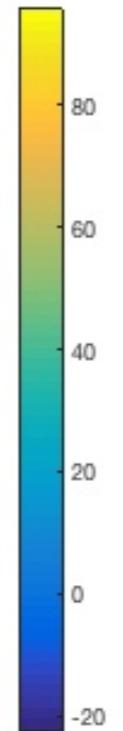
*a*



$Ku_6$



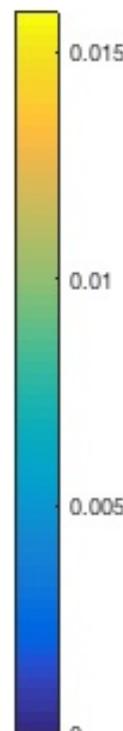
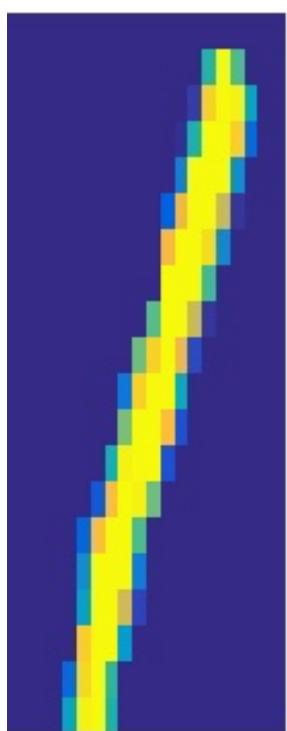
$\log(u_6)$



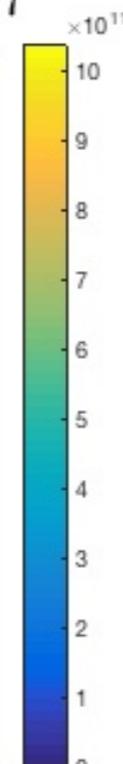
$$P_6 = D(u_6)KD(v_6)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$$

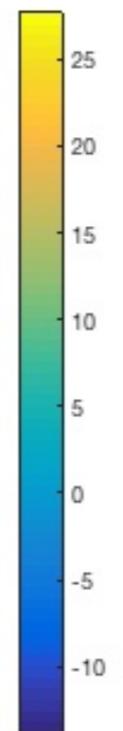
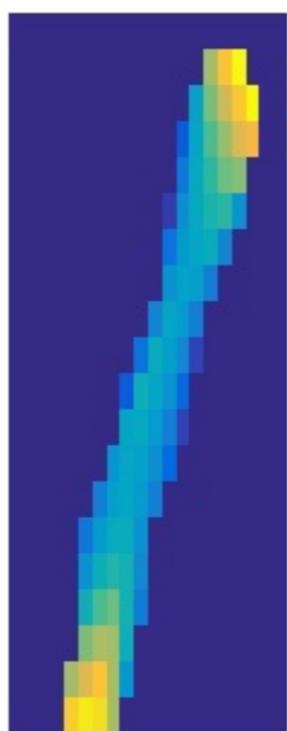
*b*



$v_7 \leftarrow b/Ku_7$

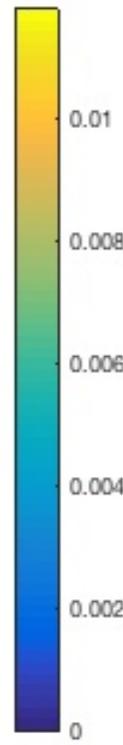
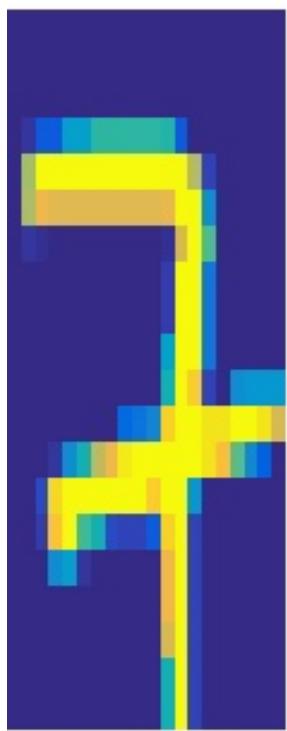


$\log(v_7)$

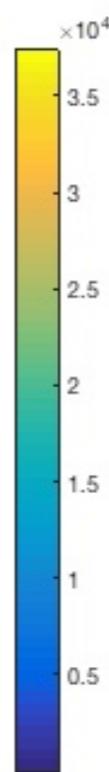


# Very Fast EMD Approx. Solver

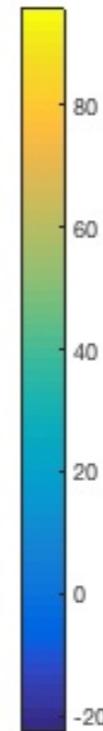
*a*



$Ku_6$



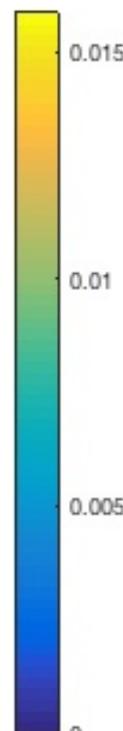
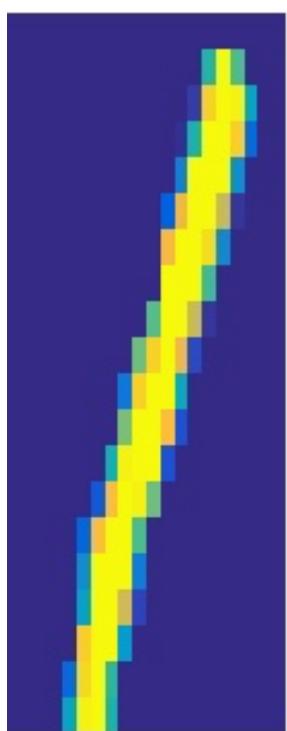
$\log(u_6)$



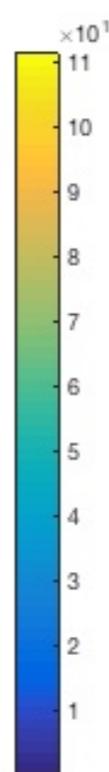
$$P_6 = D(u_6)KD(v_6)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$$

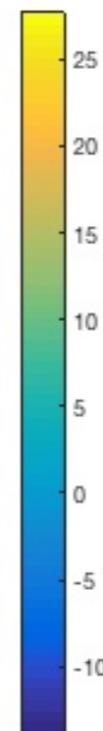
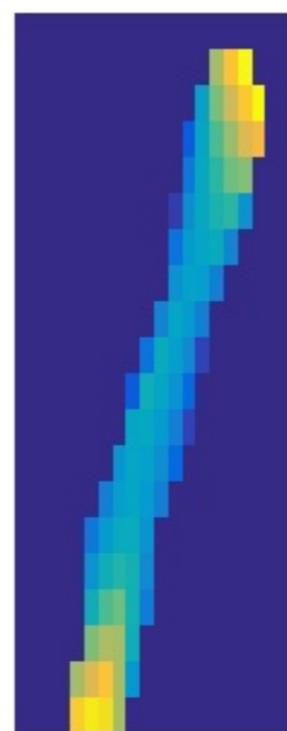
*b*



$Kv_7$

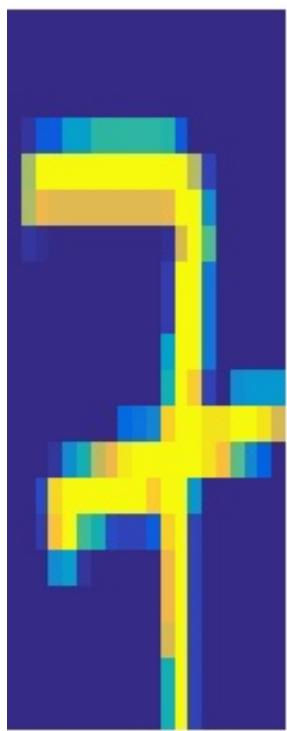


$\log(v_7)$

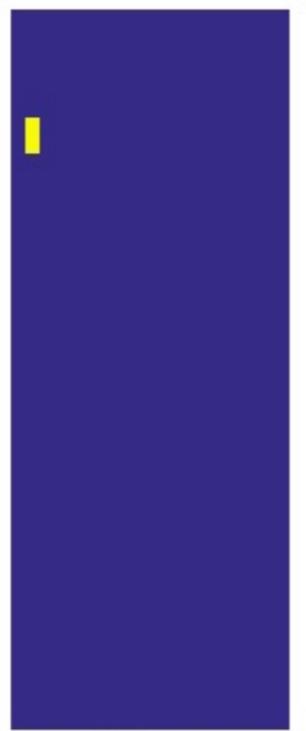


# Very Fast EMD Approx. Solver

*a*



$$u_7 \leftarrow a/Kv_7$$

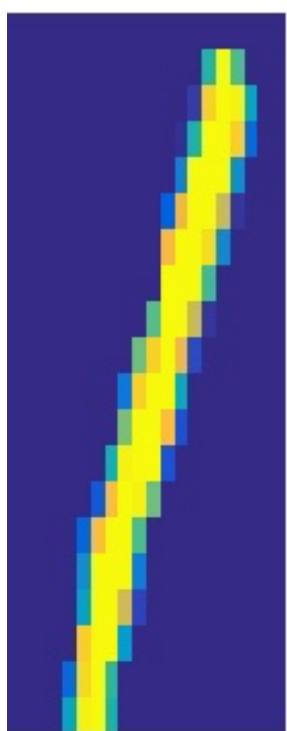


$$\log(u_7)$$

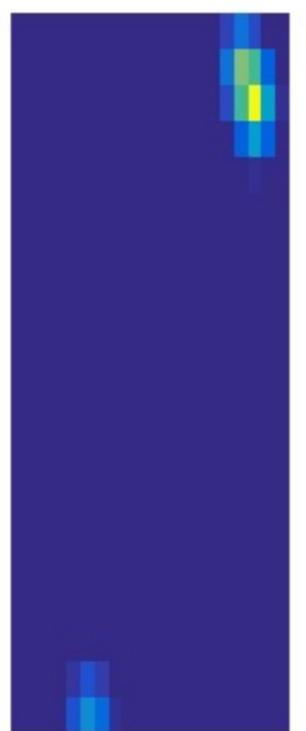


$$P_6 = D(u_6)KD(v_6)$$
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$$

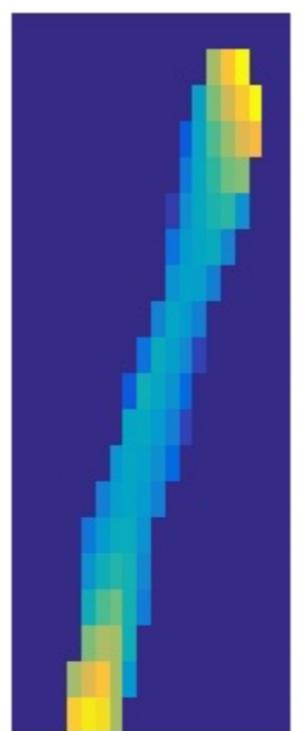
*b*



$$Kv_7$$

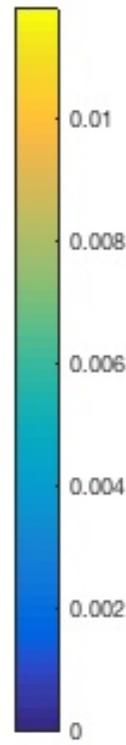
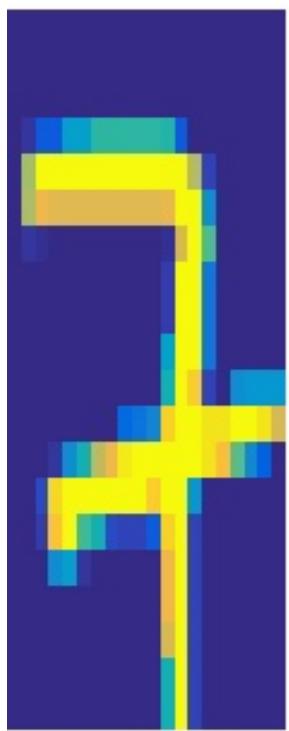


$$\log(v_7)$$

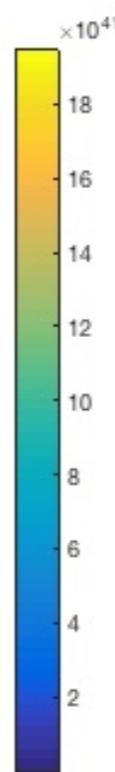


# Very Fast EMD Approx. Solver

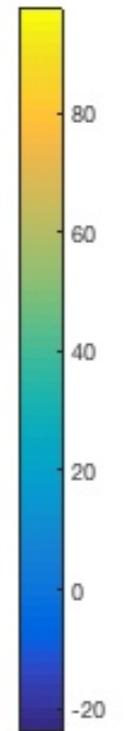
*a*



*Ku<sub>7</sub>*



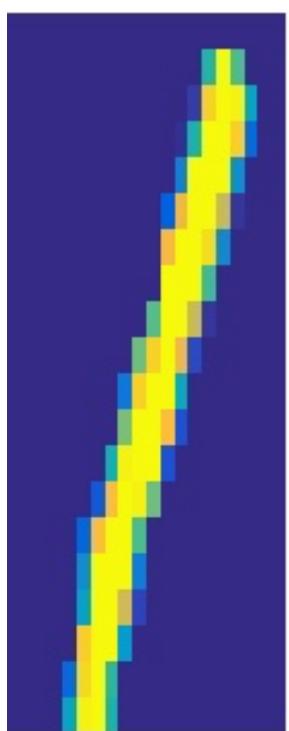
$\log(u_7)$



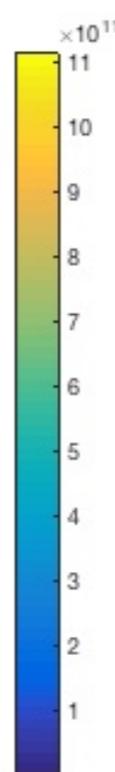
$$P_6 = D(u_6)KD(v_6)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.44948$$

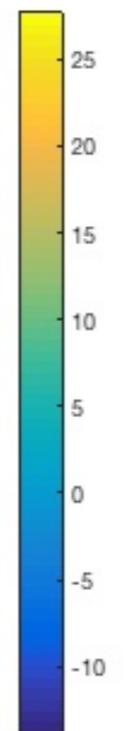
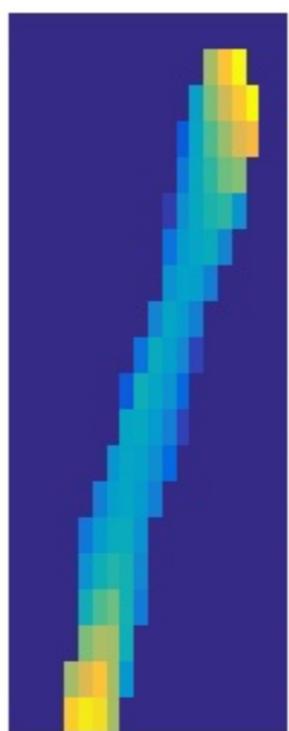
*b*



*Kv<sub>7</sub>*

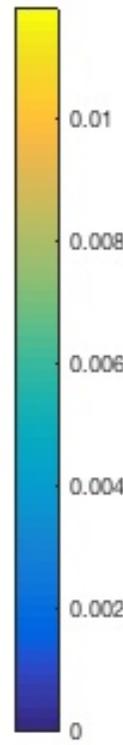
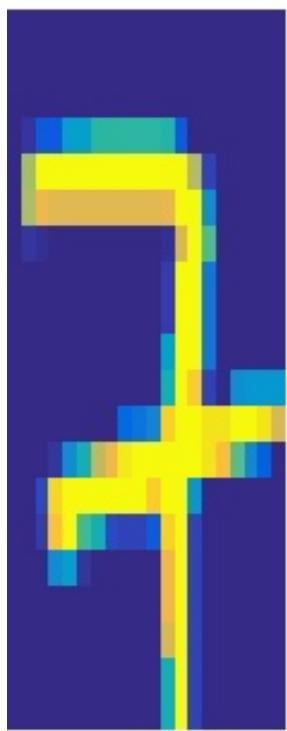


$\log(v_7)$

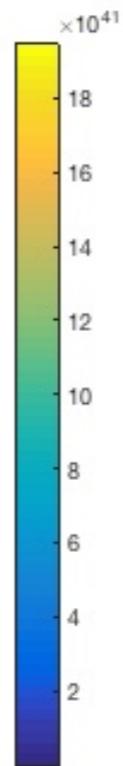


# Very Fast EMD Approx. Solver

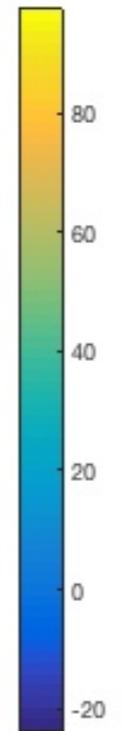
*a*



*Ku*<sub>7</sub>



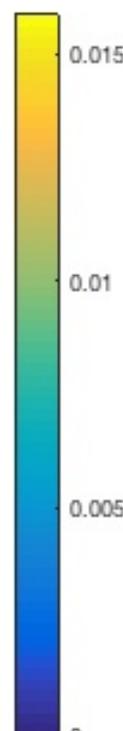
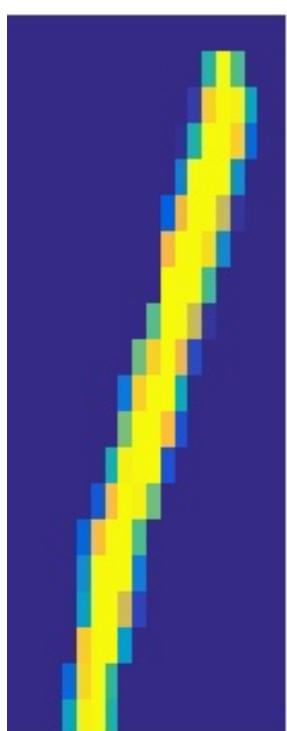
$\log(u_7)$



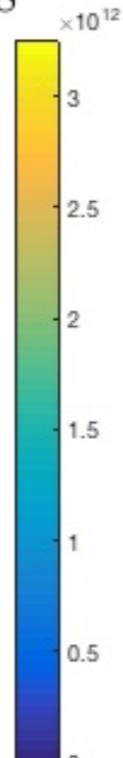
$$P_7 = D(u_7)KD(v_7)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$$

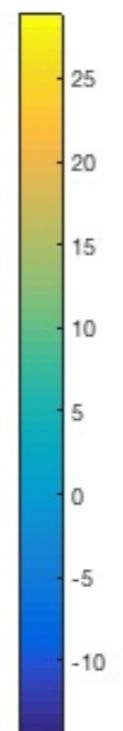
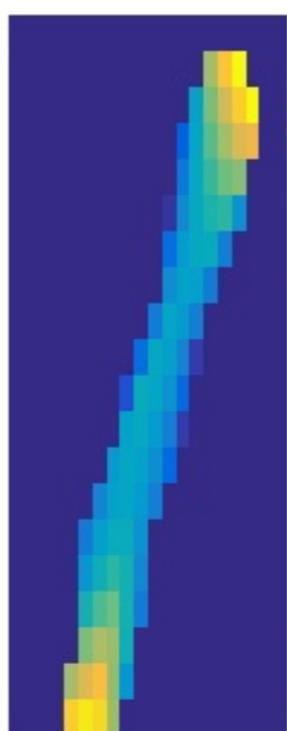
*b*



$v_8 \leftarrow b/Ku_8$

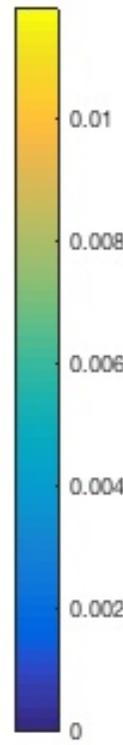
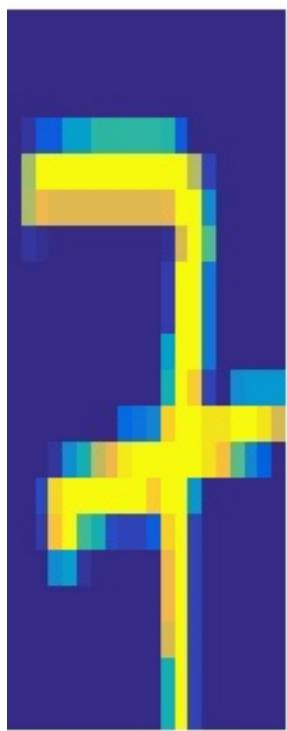


$\log(v_8)$

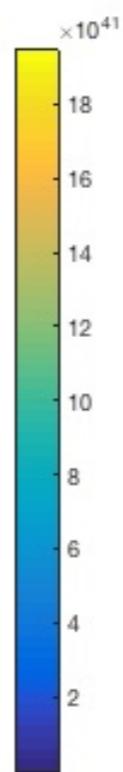


# Very Fast EMD Approx. Solver

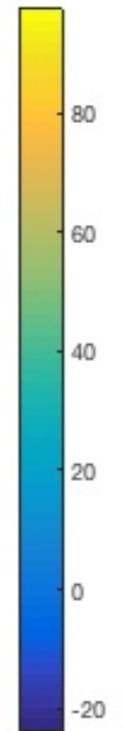
*a*



*Ku*<sub>7</sub>



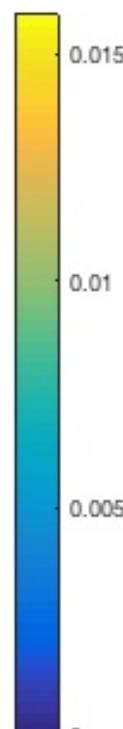
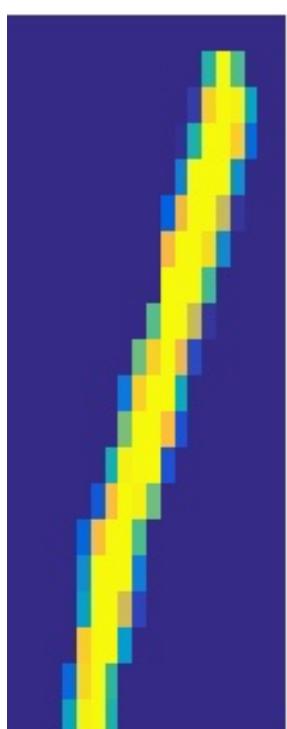
*log(u*<sub>7</sub>)



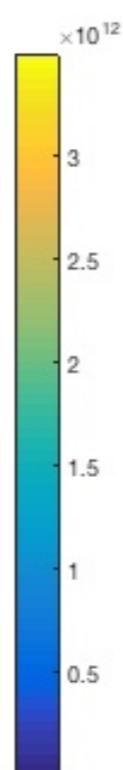
$$P_7 = D(u_7)KD(v_7)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$$

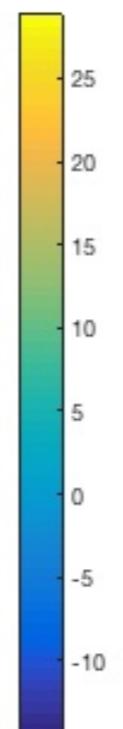
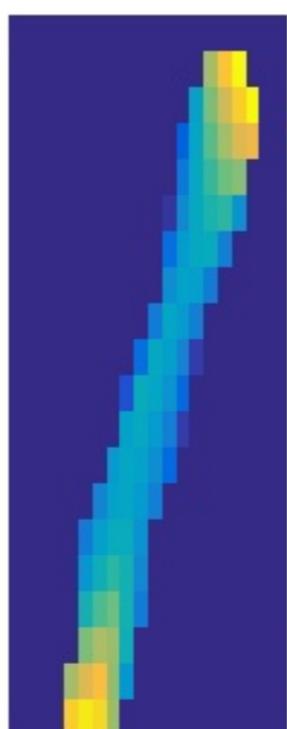
*b*



*Kv*<sub>8</sub>

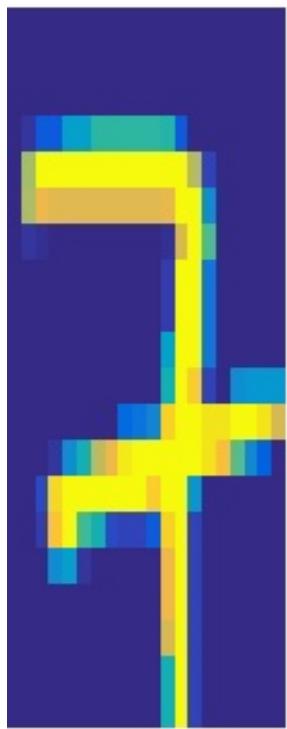


*log(v*<sub>8</sub>)



# Very Fast EMD Approx. Solver

*a*



$$u_8 \leftarrow a/Kv_8$$



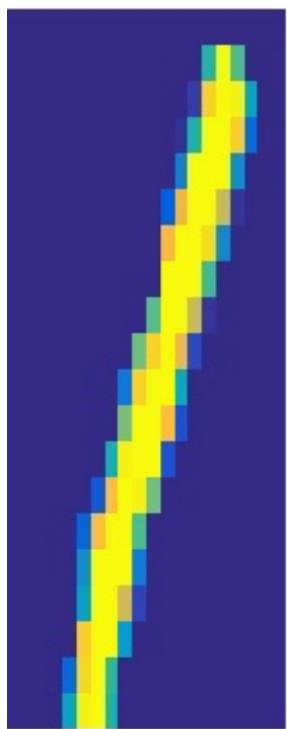
$$\log(u_8)$$



$$P_7 = D(u_7)KD(v_7)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$$

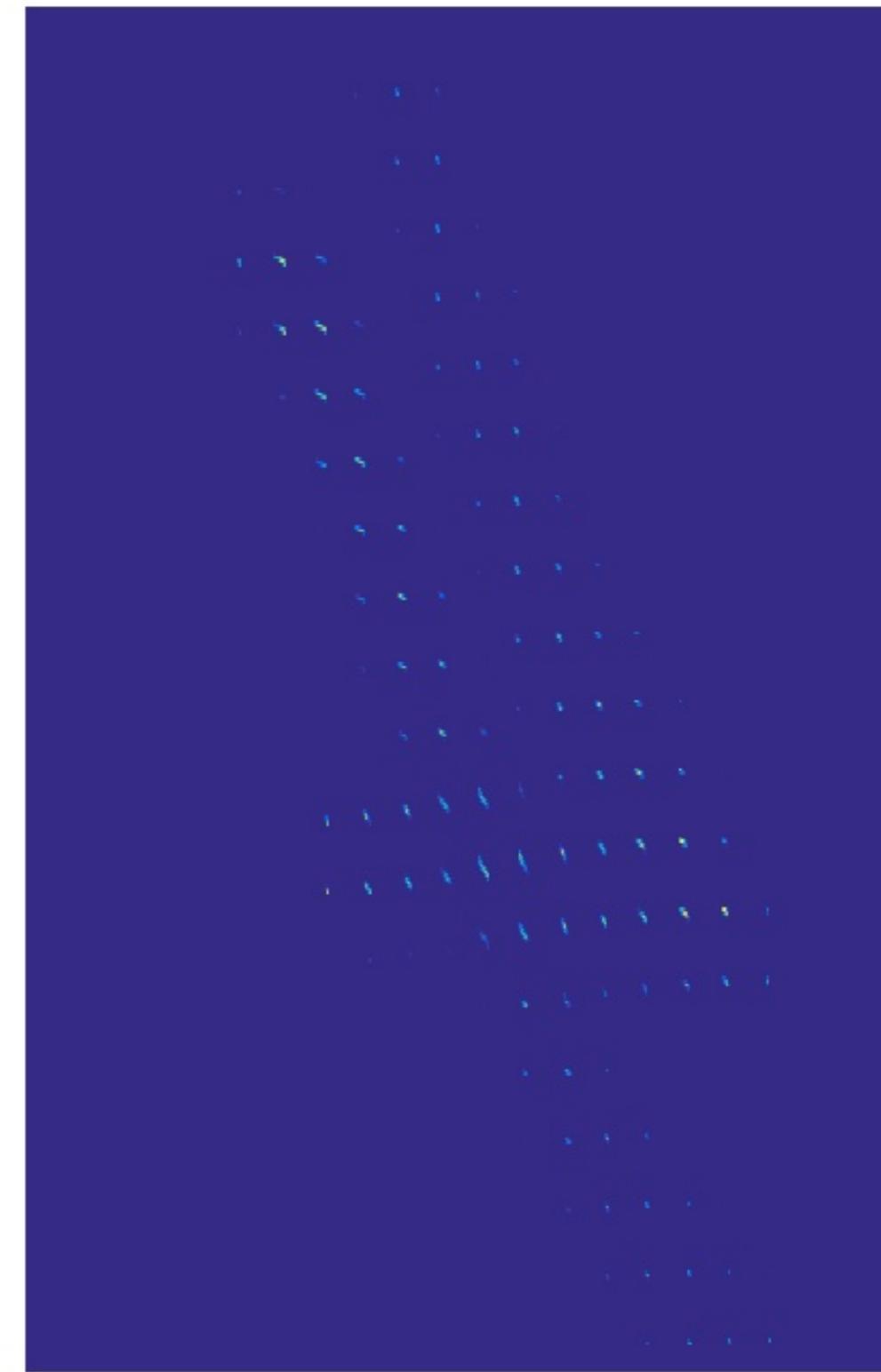
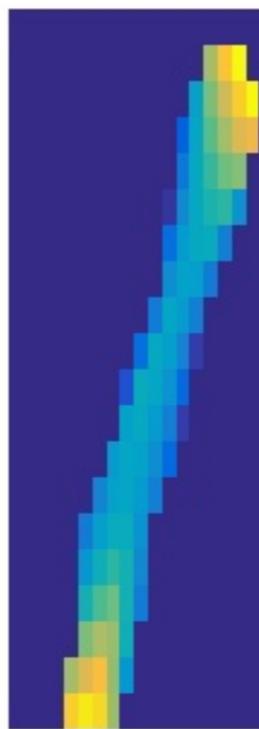
*b*



$$Kv_8$$

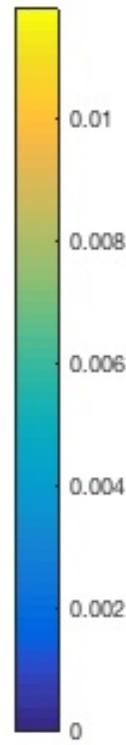
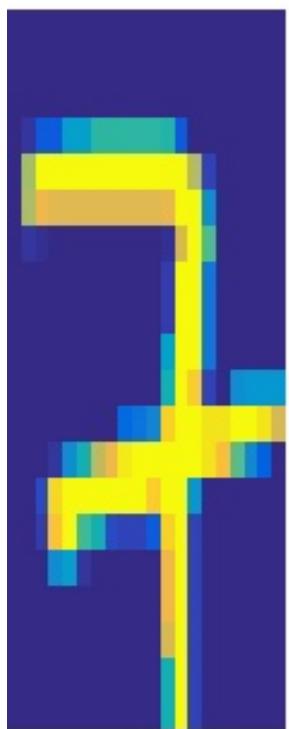


$$\log(v_8)$$

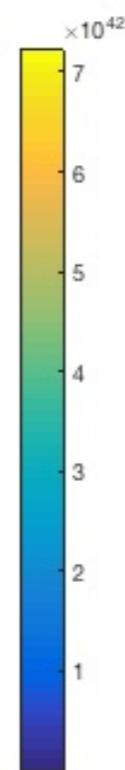


# Very Fast EMD Approx. Solver

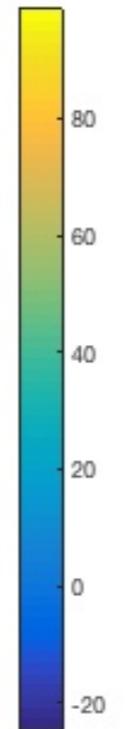
*a*



$Ku_8$



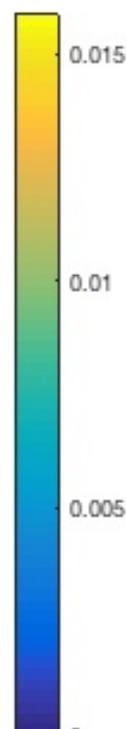
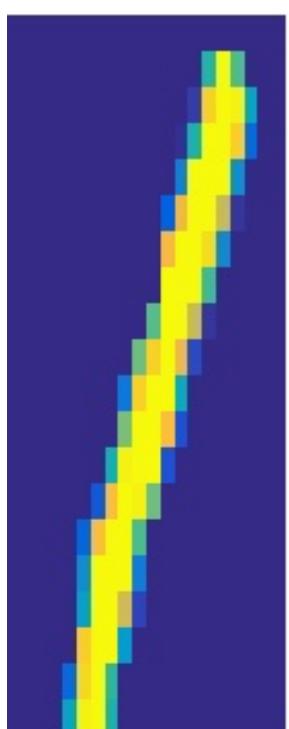
$\log(u_8)$



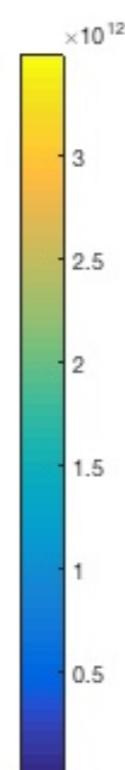
$$P_7 = D(u_7)KD(v_7)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.39738$$

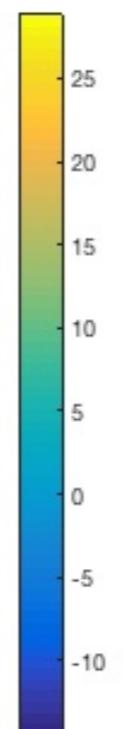
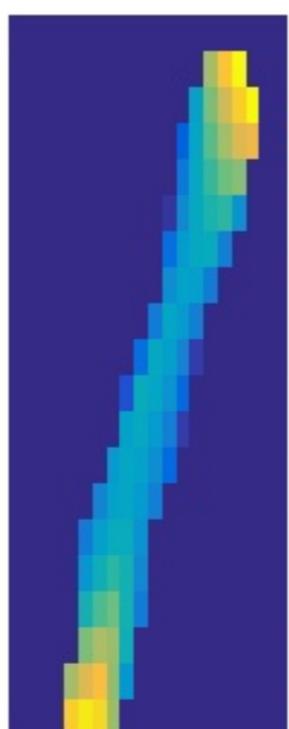
*b*



$Kv_8$

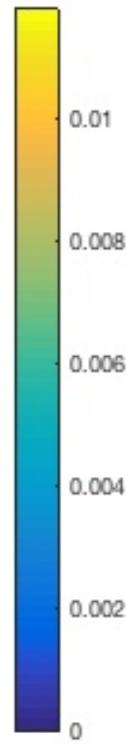
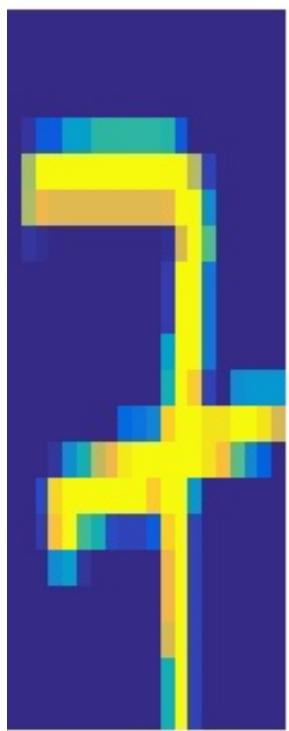


$\log(v_8)$

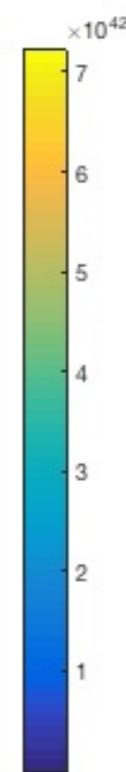


# Very Fast EMD Approx. Solver

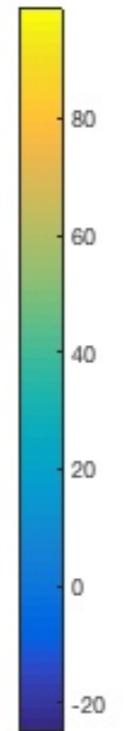
*a*



*Ku<sub>8</sub>*



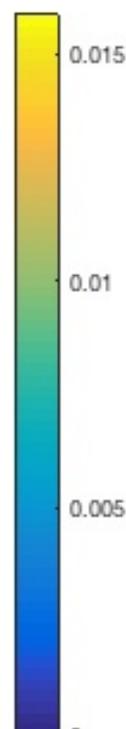
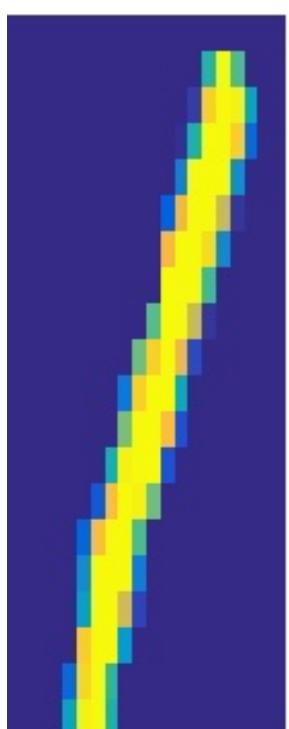
$\log(u_8)$



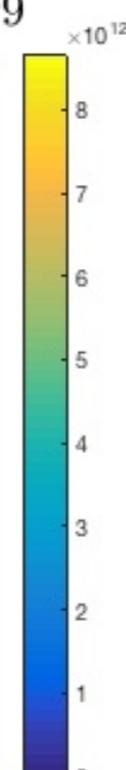
$$P_8 = D(u_8)KD(v_8)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.35442$$

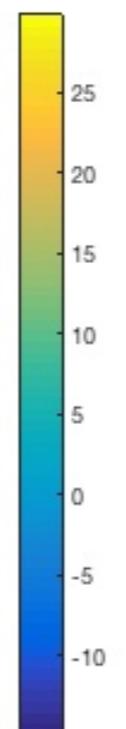
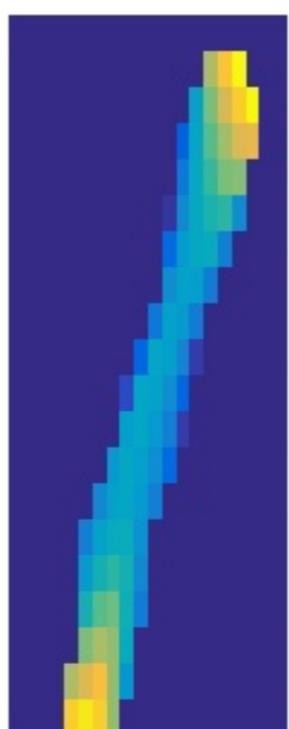
*b*



$v_9 \leftarrow b/Ku_9$

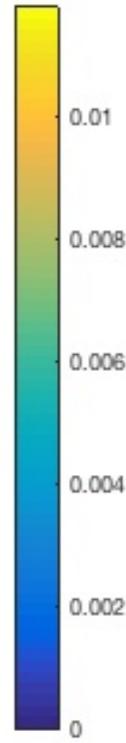
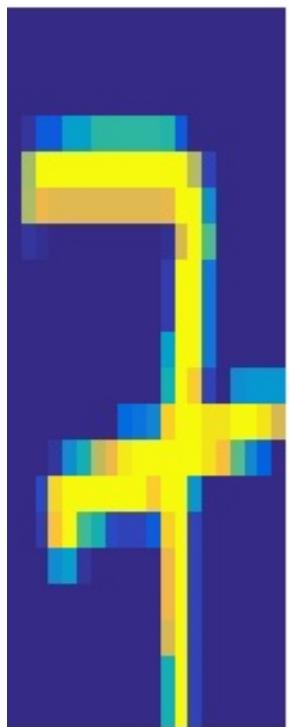


$\log(v_9)$

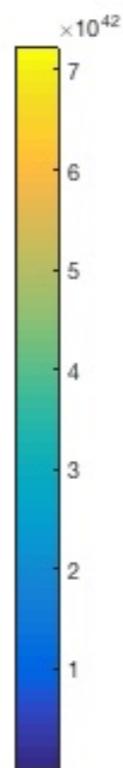


# Very Fast EMD Approx. Solver

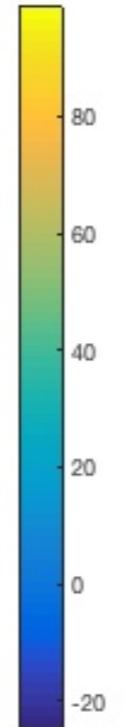
*a*



*Ku*<sub>8</sub>



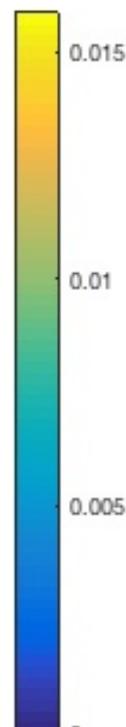
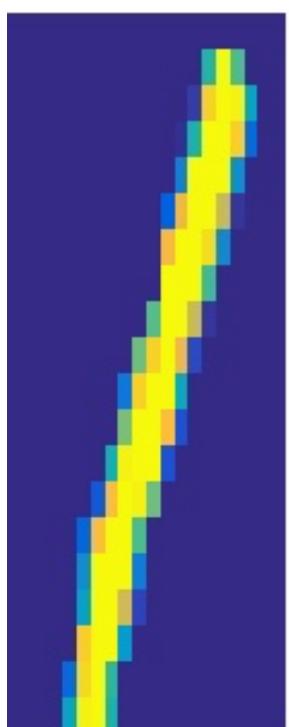
*log(u*<sub>8</sub>)



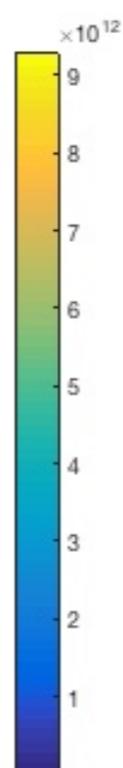
$$P_8 = D(u_8)KD(v_8)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.35442$$

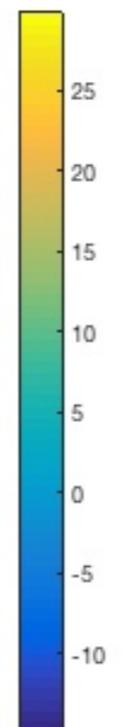
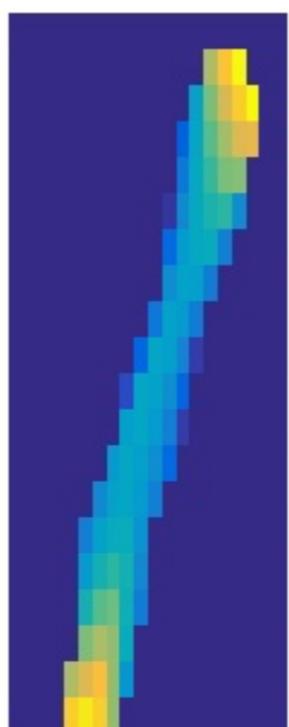
*b*



*Kv*<sub>9</sub>

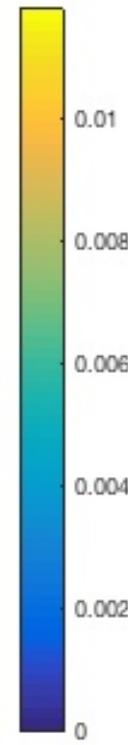
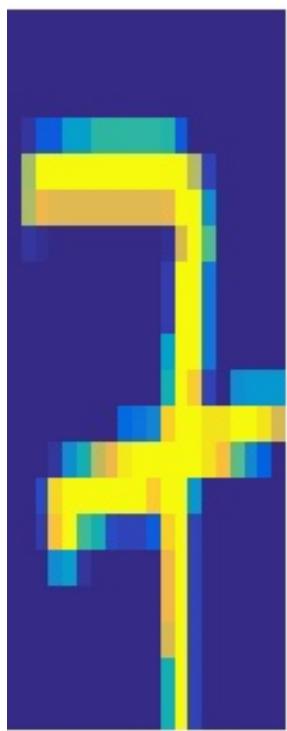


*log(v*<sub>9</sub>)



# Very Fast EMD Approx. Solver

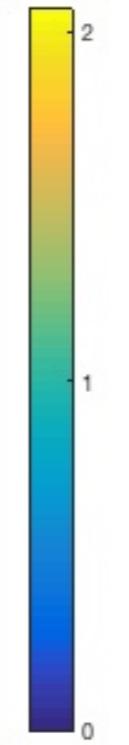
*a*



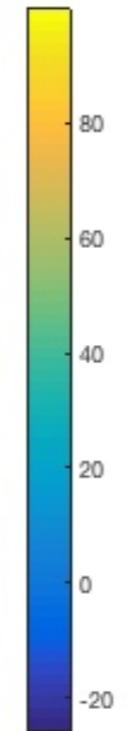
$$u_9 \leftarrow a/Kv_9$$



$\times 10^{43}$



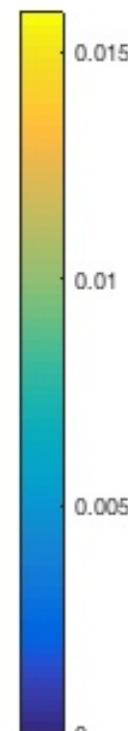
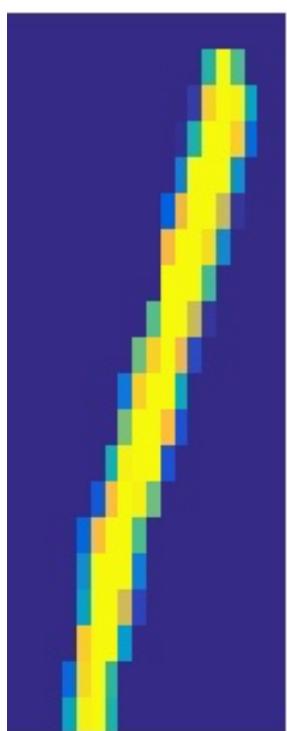
$$\log(u_9)$$



$$P_8 = D(u_8)KD(v_8)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.35442$$

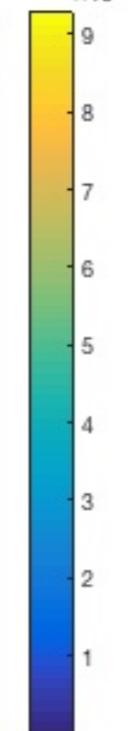
*b*



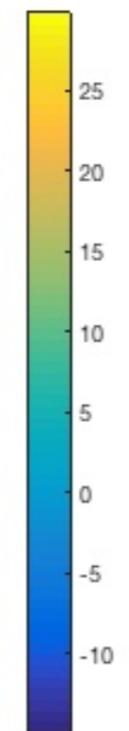
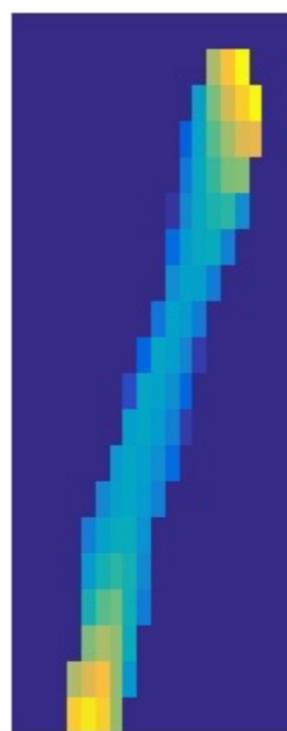
$$Kv_9$$



$\times 10^{12}$

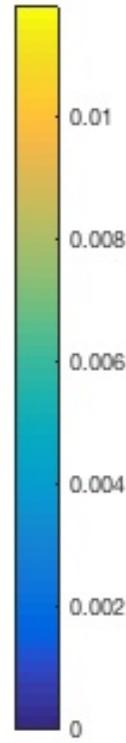
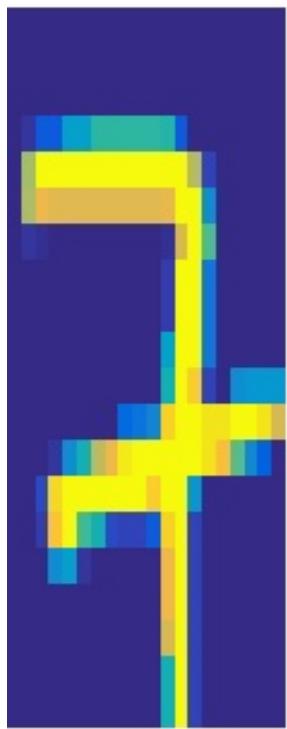


$$\log(v_9)$$

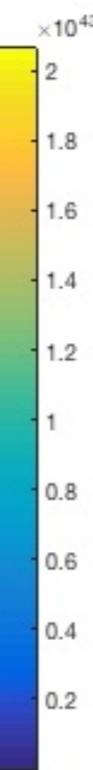


# Very Fast EMD Approx. Solver

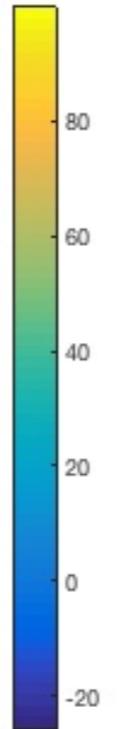
*a*



*Ku*<sub>9</sub>



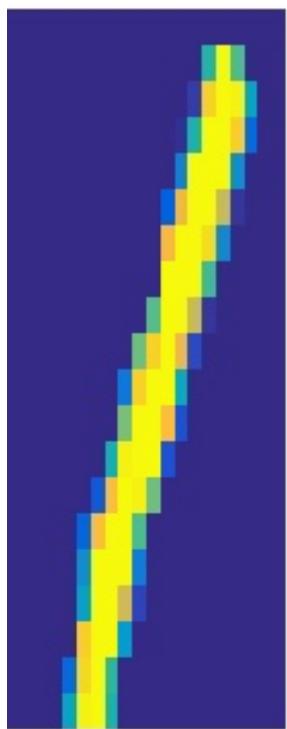
$\log(u_9)$



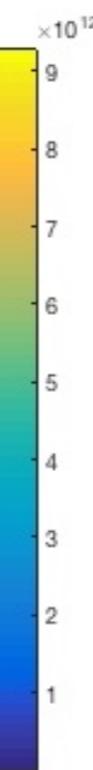
$$P_8 = D(u_8)KD(v_8)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.35442$$

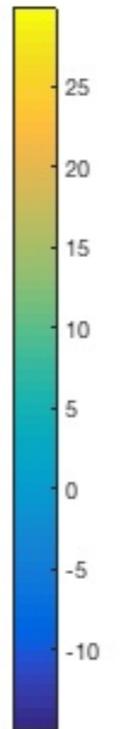
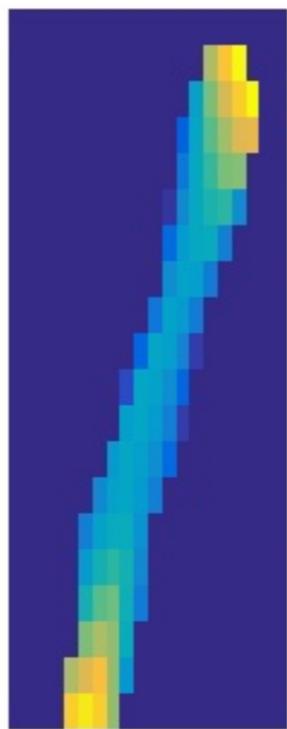
*b*



*Kv*<sub>9</sub>

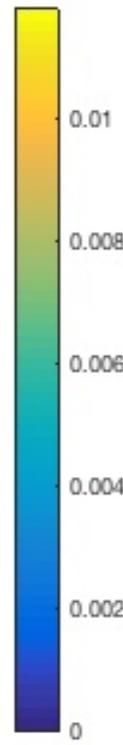
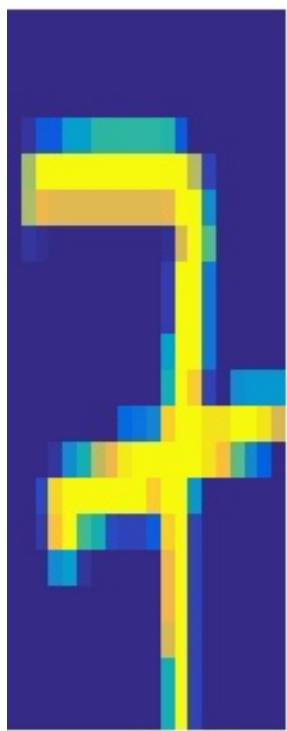


$\log(v_9)$

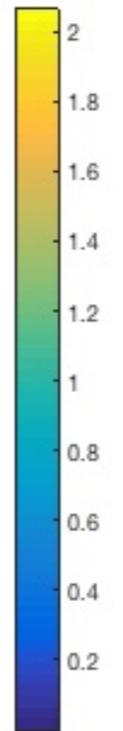


# Very Fast EMD Approx. Solver

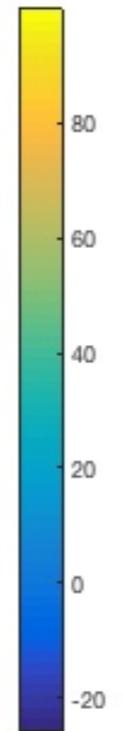
*a*



*Ku<sub>9</sub>*



$\log(u_9)$

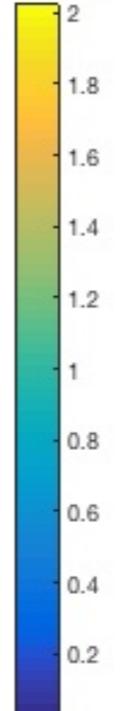
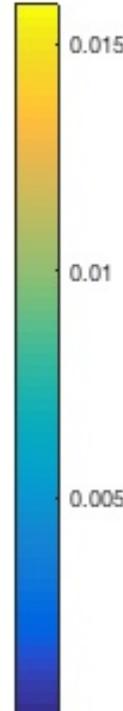
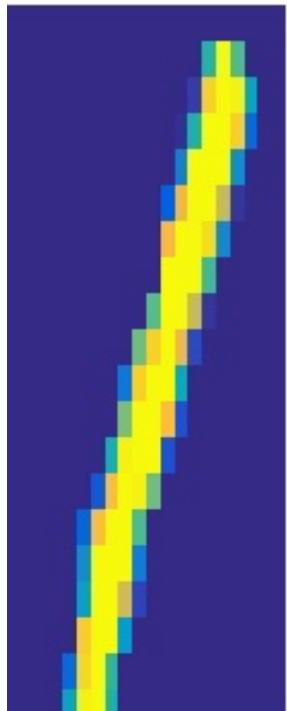


$$P_9 = D(u_9)KD(v_9)$$

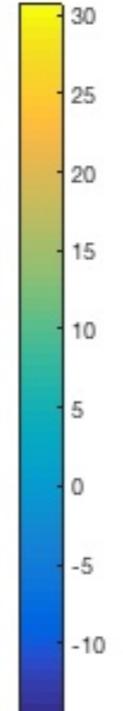
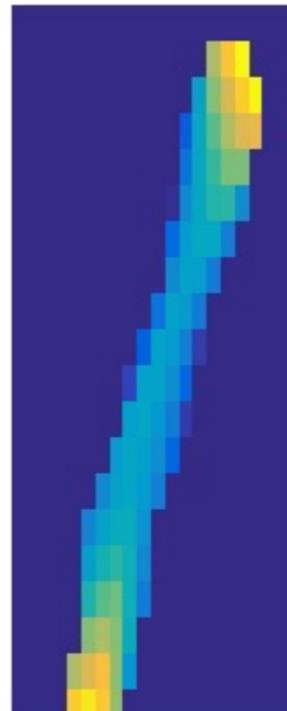
$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$$

*b*

$v_{10} \leftarrow b/Ku_{10}$

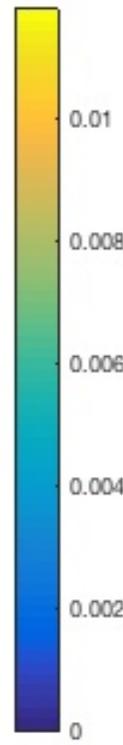
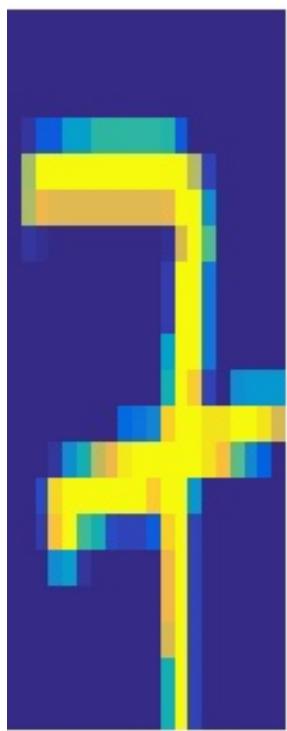


$\log(v_{10})$



# Very Fast EMD Approx. Solver

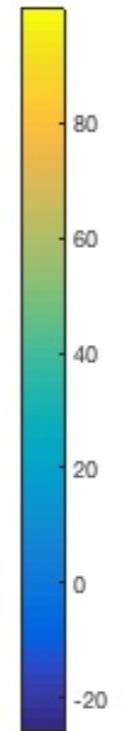
*a*



$Ku_9$



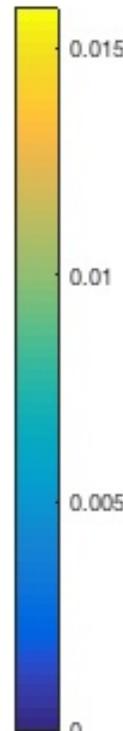
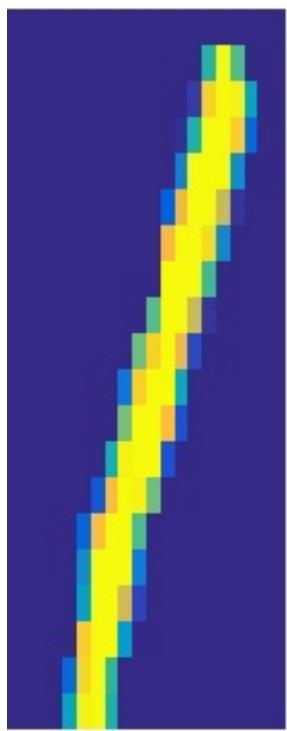
$\log(u_9)$



$$P_9 = D(u_9)KD(v_9)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$$

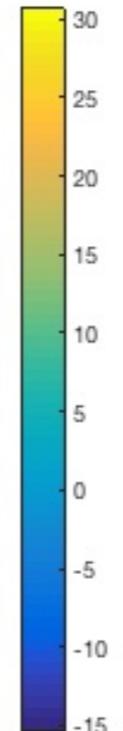
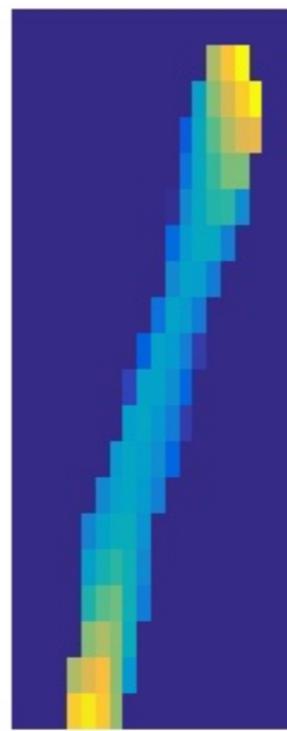
*b*



$Kv_{10}$



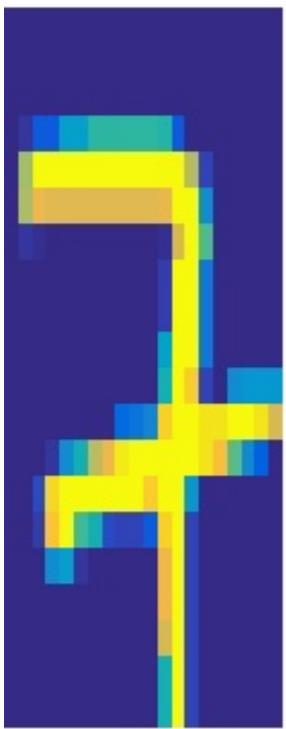
$\log(v_{10})$



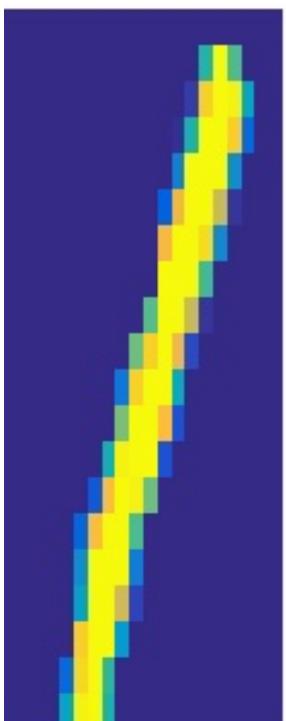
# Very Fast EMD Approx. Solver

*a*

$$u_{10} \leftarrow a/Kv_{10}$$



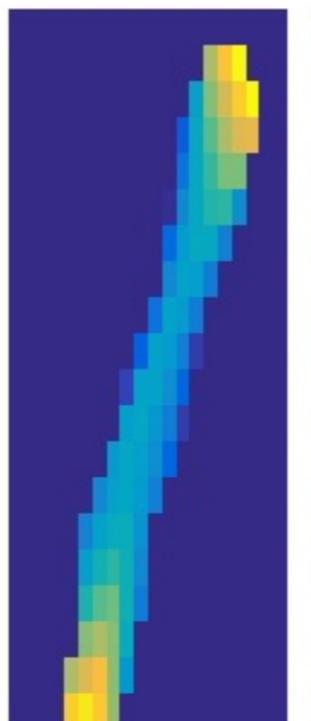
*b*



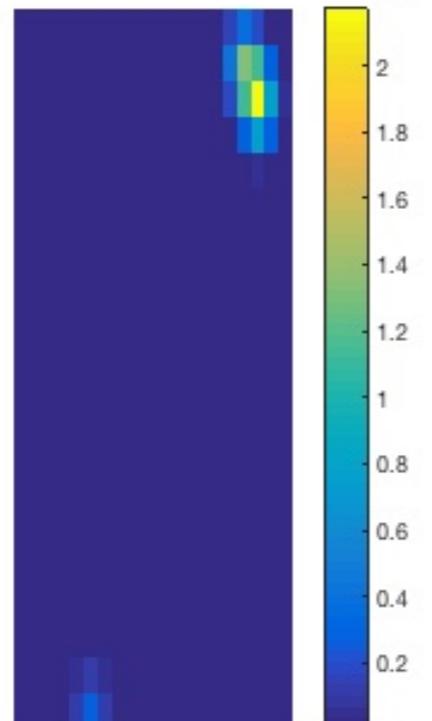
$$\log(u_{10})$$



$$\log(v_{10})$$

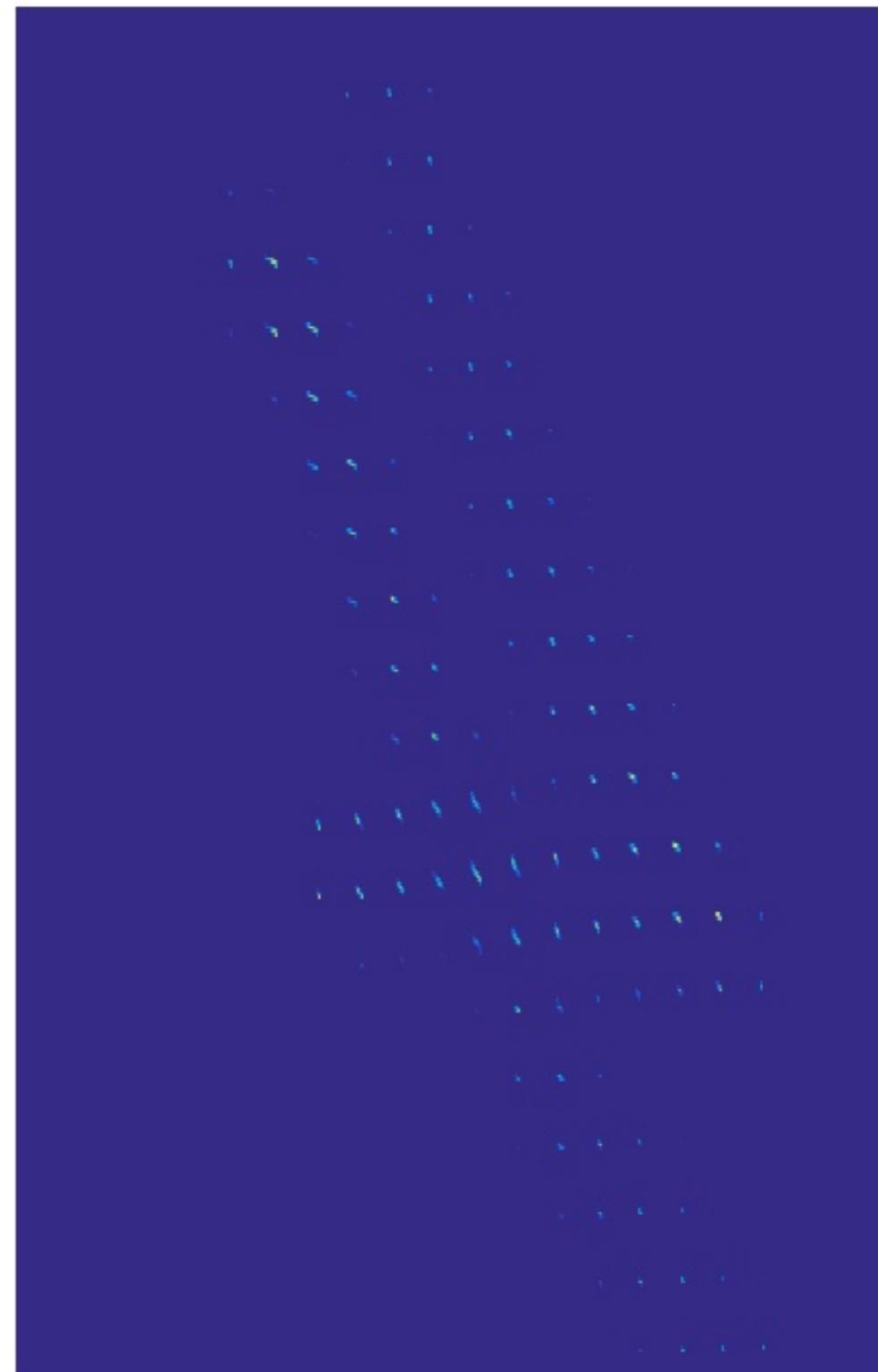


$$Kv_{10} \times 10^{13}$$



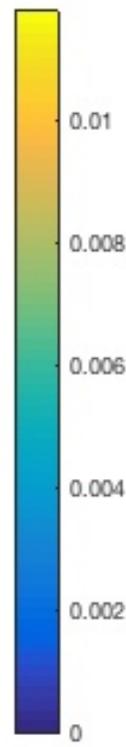
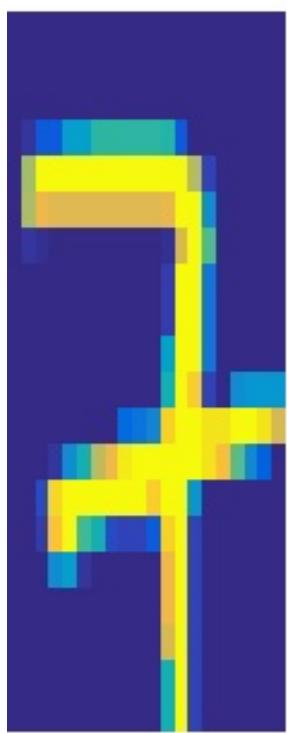
$$P_9 = D(u_9)KD(v_9)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$$



# Very Fast EMD Approx. Solver

*a*



$Ku_{10}$



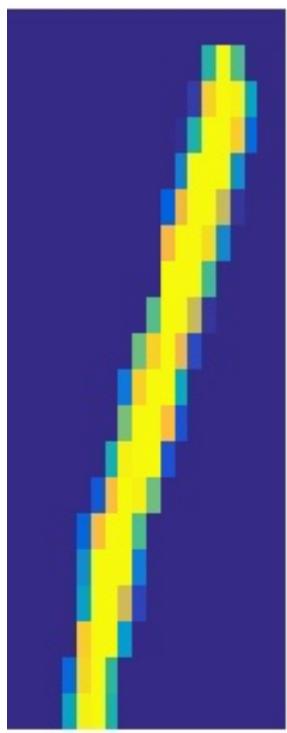
$\log(u_{10})$



$$P_9 = D(u_9)KD(v_9)$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.31916$$

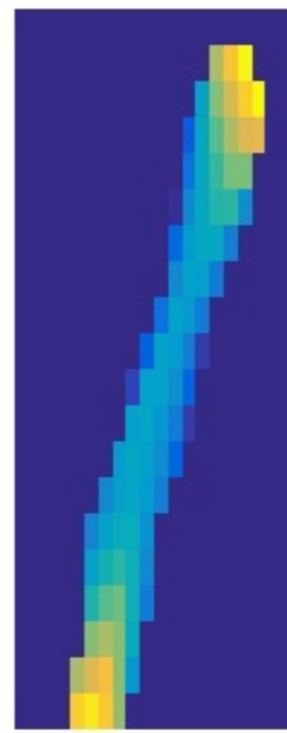
*b*



$Kv_{10}$

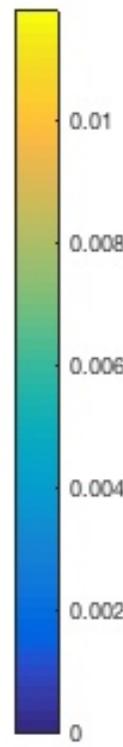
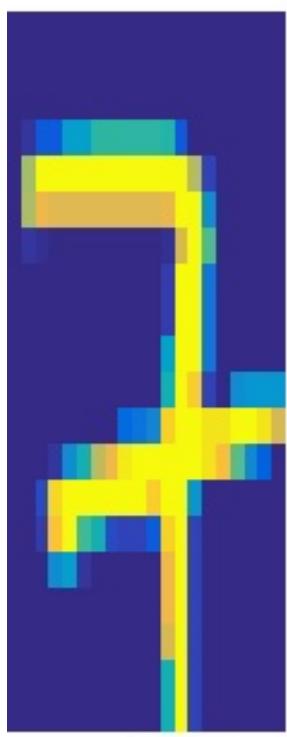


$\log(v_{10})$

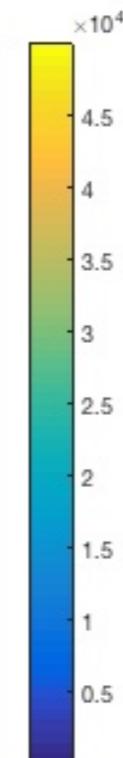


# Very Fast EMD Approx. Solver

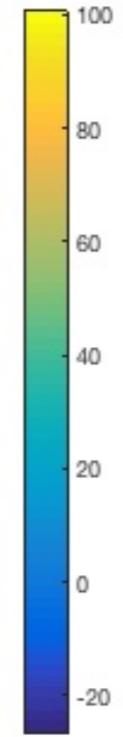
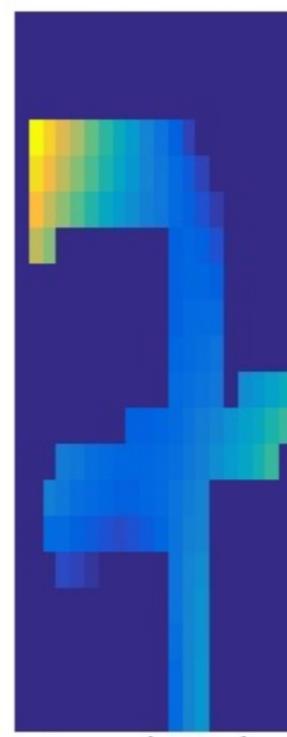
*a*



$Ku_{10}$



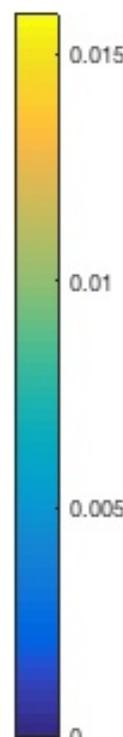
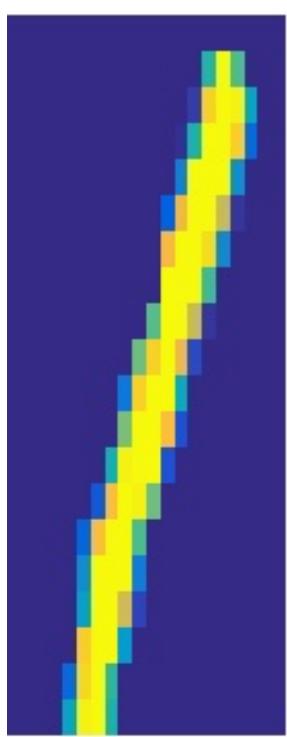
$\log(u_{10})$



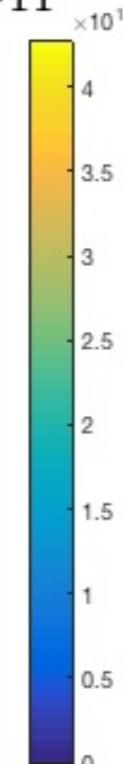
$$P_{10} = D(u_{10})KD(v_{10})$$

$$\|P1 - a\|_1 + \|P^T 1 - b\|_1 = 0.29009$$

*b*



$v_{11} \leftarrow b/Ku_{11}$



$\log(v_{11})$

