King Abdullah University of Science and Technology

KAUST

July 21-26 HONOLULU 2017

FFTLasso: Large-Scale LASSO in the Fourier Domain

Adel Bibi, Hani Itani, Bernard Ghanem

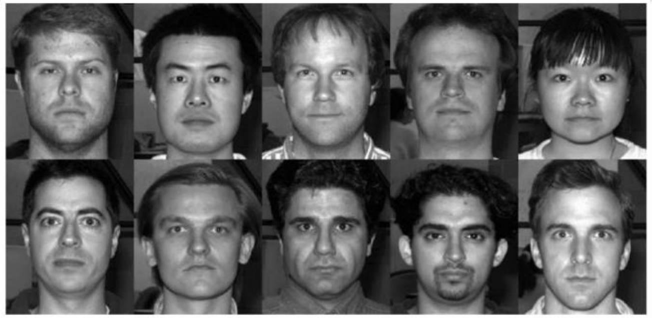
VISUAL COMPUTING CENTER

The LASSO

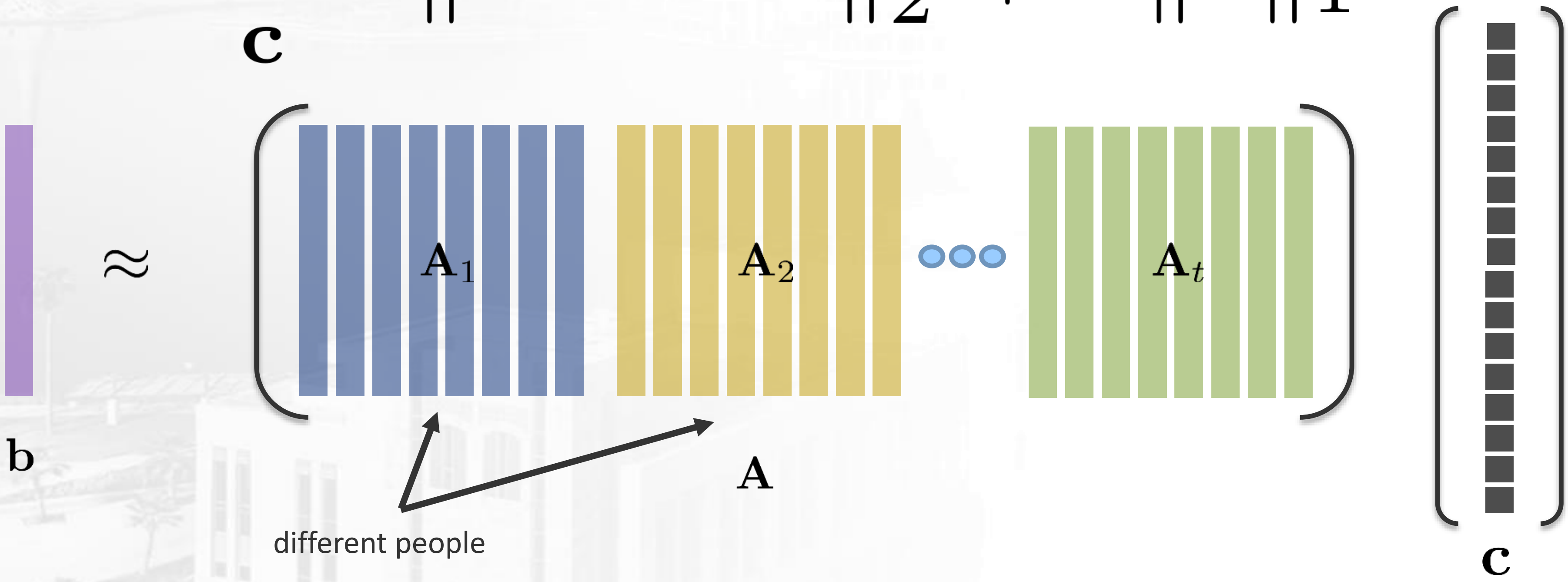
$$\min_{\mathbf{c}} \ \|\mathbf{A}\mathbf{c} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{c}\|_1$$

- The LASSO is convex but not smooth
- The matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is often fat and denoted as the dictionary
- $\mathbf{c} \in \mathbb{R}^n$ is the sparse code
- Many applications

Face Recognition



$$\min_{\mathbf{c}} \; \|\mathbf{A}\mathbf{c} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{c}\|_1$$



Solving the LASSO

Deterministic LASSO Solvers

First Order Methods

Pros: Inexpensive iteration cost
Cons: Lower convergence rate

Examples

(1) Proximal point methods
(2) ADMMs

Second Order Methods

Pros: Quadratic convergence rates
Cons: Expensive Iteration cost

Examples

Primal dual interior point methods

Solving the LASSO

The Focus of this work is on Deterministic First Order Methods

Deterministic LASSO Solvers

- **First Order Methods**
  - **Pros:** Inexpensive iteration cost
  - **Cons:** Lower convergence rate
  - **Examples**
    - (1) Proximal point methods
    - (2) ADMMs
- **Second Order Methods**
  - **Pros:** Quadratic convergence rates
  - **Cons:** Expensive Iteration cost
  - **Examples**
    - Primal dual interior point methods

ADMM

$$\min_{\mathbf{c}} \; \|\mathbf{A}\mathbf{c} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{c}\|_1$$

ADMM

Primal Lasso (PL-ADMM)

$$\boxed{(2\mathbf{A}^\top\mathbf{A} + u_k\mathbf{I}_n)\mathbf{c}_{k+1} = 2\mathbf{A}^\top\mathbf{b} - \Psi_k + u_k\mathbf{z}_k}$$

Solving a PD linear system of size $n \times n$

Dual Lasso (DL-ADMM)

$$\boxed{(\rho_k\mathbf{A}\mathbf{A}^\top + \tfrac{1}{2}\mathbf{I}_m)\Psi_{k+1} = \mathbf{A}(\rho_k\zeta - \mathbf{c}) - \mathbf{b}}$$

Solving a PD linear system of size $m \times m$

$$\mathbf{A} \in \mathbb{R}^{m\times n}$$

DL-ADMM

Pros:

- Empirically shown to be one of the fastest first order deterministic LASSO solver
- Handles Linear Constraints

DL-ADMM

Pros:

- Empirically shown to be one of the fastest first order deterministic LASSO solver
- Handles Linear Constraints

Cons:

- Requires linear system solver routine
- Not trivially parallelized over many GPUs
- Scalability issues

New Solver?

The solver we seek should enjoy:

Low iteration complexity

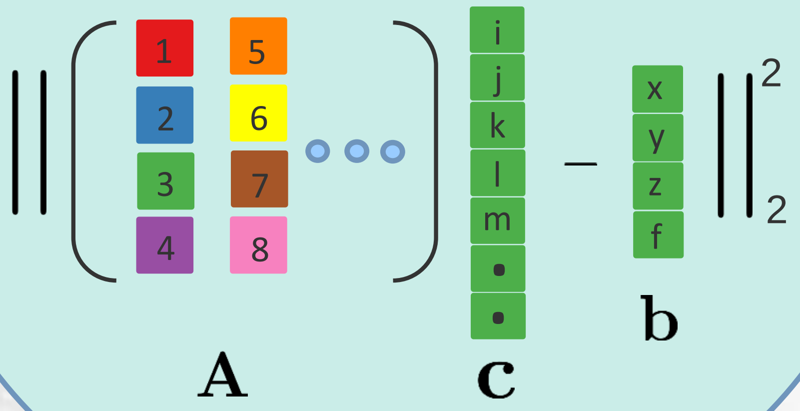
ADMM’s linear convergence rate

Trivially distributed over multiple GPUs

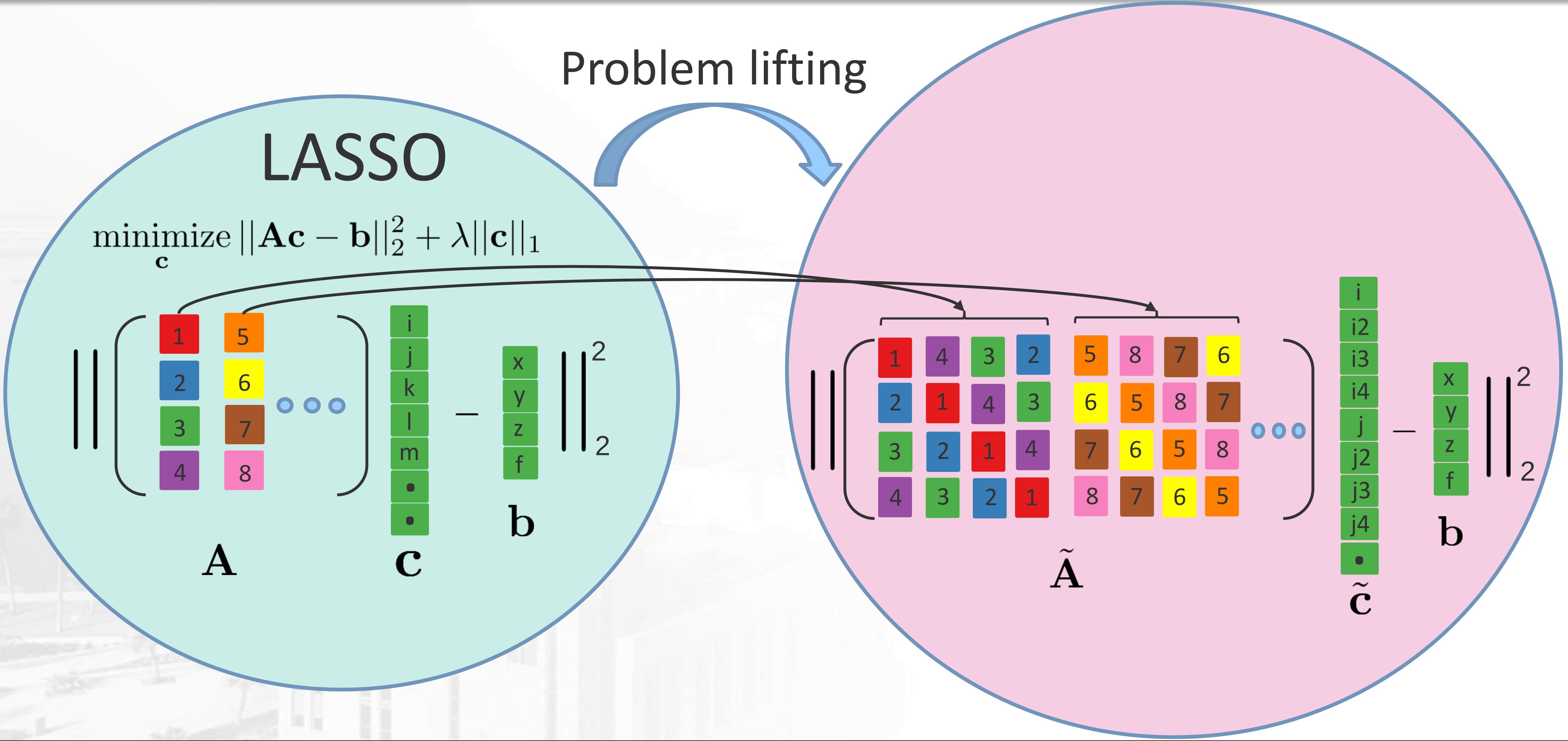
Problem Lifting

LASSO

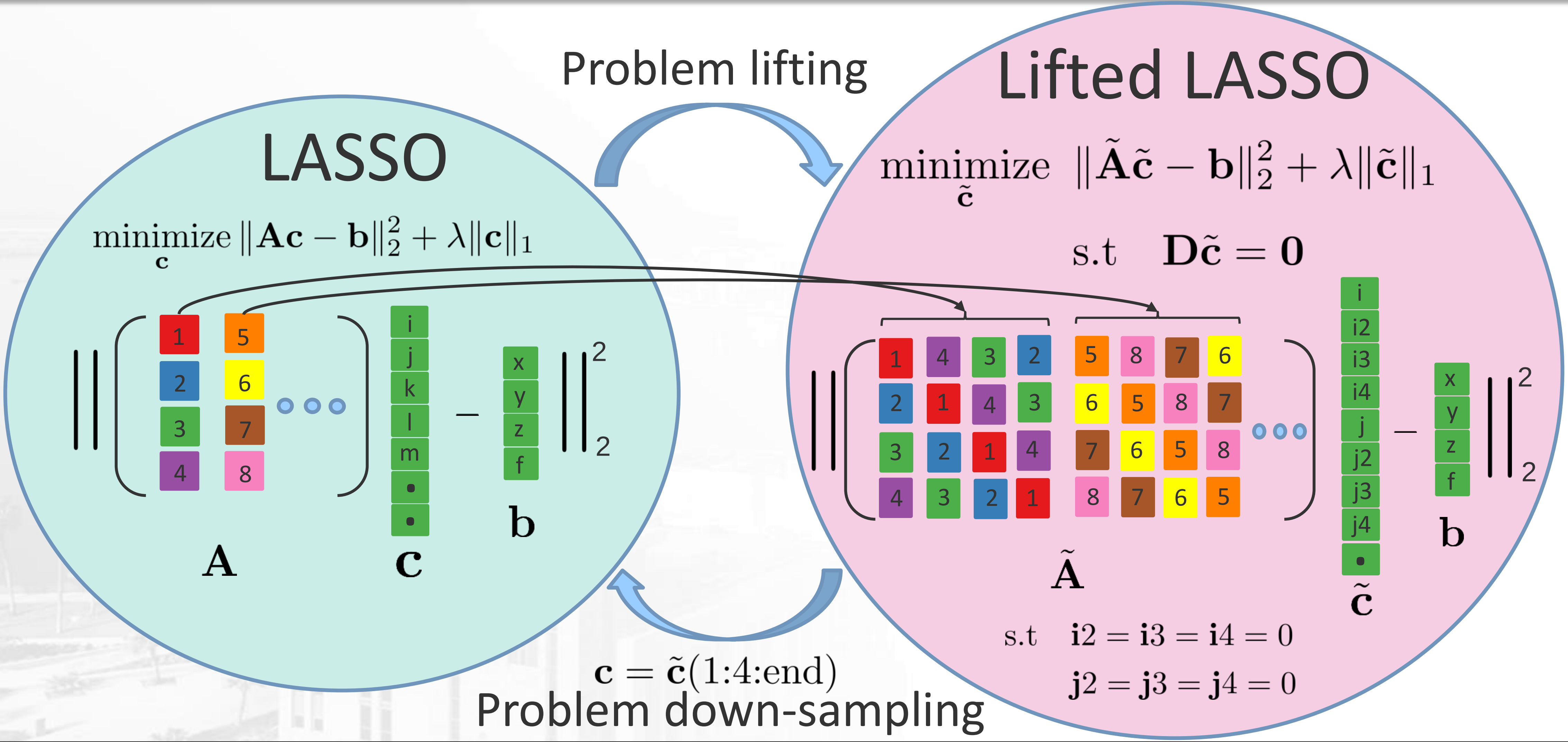
$$\underset{\mathbf{c}}{\text{minimize}}\ ||\mathbf{A}\mathbf{c} - \mathbf{b}||_2^2 + \lambda||\mathbf{c}||_1$$



Problem Lifting



Problem Lifting



Lifted Problem as Constrained CSC

$$\min_{\tilde{\mathbf{c}}} \quad \left\|\tilde{\mathbf{A}}\tilde{\mathbf{c}} - \mathbf{b}\right\|_2^2 + \lambda \left\|\tilde{\mathbf{c}}\right\|_1 \quad \text{s.t} \quad \mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$$

Lifted Problem as Constrained CSC

$$\min_{\tilde{\mathbf{c}}} \quad \left\| \tilde{\mathbf{A}}\tilde{\mathbf{c}} - \mathbf{b} \right\|_2^2 + \lambda \left\| \tilde{\mathbf{c}} \right\|_1 \quad \text{s.t} \quad \mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$$

$$\Updownarrow$$

$$\min_{\tilde{\mathbf{c}}} \left\| \sum_{i}^{n} \mathbf{a}_i * \tilde{\mathbf{c}}_i - \mathbf{b} \right\|_2^2 + \lambda \left\| \tilde{\mathbf{c}} \right\|_1 \quad \text{s.t} \quad \mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$$

Lifted Problem as Constrained CSC

$$\min_{\tilde{\mathbf{c}}} \quad \left\|\tilde{\mathbf{A}}\tilde{\mathbf{c}} - \mathbf{b}\right\|_2^2 + \lambda \left\|\tilde{\mathbf{c}}\right\|_1 \quad \text{s.t} \quad \mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$$

$$\Updownarrow$$

$$\min_{\tilde{\mathbf{c}}} \left\|\sum_i^n \mathbf{a}_i * \tilde{\mathbf{c}}_i - \mathbf{b}\right\|_2^2 + \lambda \left\|\tilde{\mathbf{c}}\right\|_1 \quad \text{s.t} \quad \mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$$

$$\Updownarrow$$

Since $\mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$ then $\tilde{\mathbf{c}}_i^\top = \left[c_i,\ \mathbf{0}_{m-1}^\top\right]^\top \forall i$

$$\min_{\tilde{\mathbf{c}}} \left\|\sum_i^n \mathbf{a}_i c_i - \mathbf{b}\right\|_2^2 + \lambda \|\tilde{\mathbf{c}}\|_1$$

Lifted Problem as Constrained CSC

$$\min_{\tilde{\mathbf{c}}} \left\| \tilde{\mathbf{A}}\tilde{\mathbf{c}} - \mathbf{b} \right\|_2^2 + \lambda \left\| \tilde{\mathbf{c}} \right\|_1 \quad \text{s.t} \quad \mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$$

$$\Updownarrow$$

$$\min_{\tilde{\mathbf{c}}} \left\| \sum_i^n \mathbf{a}_i * \tilde{\mathbf{c}}_i - \mathbf{b} \right\|_2^2 + \lambda \left\| \tilde{\mathbf{c}} \right\|_1 \quad \text{s.t} \quad \mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$$

$$\Updownarrow$$

Since $\mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$ then $\tilde{\mathbf{c}}_i^\top = \left[c_i, \ \mathbf{0}_{m-1}^\top\right]^\top \forall i$

$$\min_{\tilde{\mathbf{c}}} \left\| \sum_i^n \mathbf{a}_i c_i - \mathbf{b} \right\|_2^2 + \lambda \|\tilde{\mathbf{c}}\|_1$$

$$\Updownarrow$$

Since $\|\tilde{\mathbf{c}}\|_1 = \|\mathbf{c}\|_1$

$$\min_{\mathbf{c}} \|\mathbf{A}\mathbf{c} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{c}\|_1$$

The Big Picture

Problem lifting

DL-ADMM

$$\min_{\mathbf{c}} \ ||\mathbf{Ac} - \mathbf{b}||_2^2 + \lambda||\mathbf{c}||_1$$

$$\min_{\Psi} \ \frac{1}{4}||\Psi||_2^2 + \Psi^\top \mathbf{b}$$

$$\text{s.t.} \quad ||\mathbf{A^H}\Psi||_\infty \leq \lambda$$

FFTLasso

$$\min_{\tilde{\mathbf{c}}} \ ||\tilde{\mathbf{A}}\tilde{\mathbf{c}} - \mathbf{b}||_2^2 + \lambda||\tilde{\mathbf{c}}||_1$$

$$\text{s.t} \quad \mathbf{D}\tilde{\mathbf{c}} = \mathbf{0}$$

$$\min_{\Psi,\theta} \ \frac{1}{4}||\Psi||_2^2 + \Psi^{\mathbf{H}}\mathbf{b}$$

$$\text{s.t} \ ||\tilde{\mathbf{A}}^{\mathbf{H}}\Psi + \mathbf{D^H}\theta||_\infty \leq \lambda$$

FFTLasso

FFTLasso

$$\min_{\Psi,\theta} \frac{1}{4}\|\Psi\|_2^2 + \Psi^{\mathbf{H}}\mathbf{b} \quad \text{s.t} \quad \left\|\tilde{\mathbf{A}}^{\mathbf{H}}\Psi + \mathbf{D}^{\mathbf{H}}\theta\right\|_\infty \leq \lambda$$

Update $\hat{\Psi}^*$:

$$(\rho\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathbf{H}} + \frac{1}{2}\mathbf{I}_m)\Psi = \tilde{\mathbf{A}}(\rho\zeta - \rho\mathbf{D}^{\mathbf{H}}\theta - \tilde{\mathbf{c}}) - \mathbf{b}$$

$$\hat{\Psi}^* = \frac{\sum_i^N \hat{\mathbf{a}}_i^* \odot \hat{\mathbf{e}}_i^* - \hat{\mathbf{b}}^*}{\rho\sum_i^N \hat{\mathbf{a}}_i \odot \hat{\mathbf{a}}_i^* + \frac{1}{2}\mathbf{1}_m}$$

where $\quad \mathbf{e} = \rho\zeta - \rho\mathbf{D}^{\mathbf{H}}\theta - \tilde{\mathbf{c}} = \left[\mathbf{e}_1^{\mathbf{H}}, \ldots, \mathbf{e}_n^{\mathbf{H}}\right]^{\mathbf{H}}$

FFTLasso

FFTLasso

$$\min_{\Psi,\theta} \frac{1}{4}\|\Psi\|_2^2 + \Psi^{\mathbf{H}}\mathbf{b} \quad \text{s.t} \quad \left\|\tilde{\mathbf{A}}^{\mathbf{H}}\Psi + \mathbf{D}^{\mathbf{H}}\theta\right\|_\infty \leq \lambda$$

Update $\hat{\Psi}^*$:

$$(\rho\tilde{\mathbf{A}}\tilde{\mathbf{A}}^{\mathbf{H}} + \frac{1}{2}\mathbf{I}_m)\Psi = \tilde{\mathbf{A}}(\rho\zeta - \rho\mathbf{D}^{\mathbf{H}}\theta - \tilde{\mathbf{c}}) - \mathbf{b}$$

$$\hat{\Psi}^* = \frac{\sum_i^N \hat{\mathbf{a}}_i^* \odot \hat{\mathbf{e}}_i^* - \hat{\mathbf{b}}^*}{\rho\sum_i^N \hat{\mathbf{a}}_i \odot \hat{\mathbf{a}}_i^* + \frac{1}{2}\mathbf{1}_m}$$

where $\mathbf{e} = \rho\zeta - \rho\mathbf{D}^{\mathbf{H}}\theta - \tilde{\mathbf{c}} = \left[\mathbf{e}_1^{\mathbf{H}}, \dots, \mathbf{e}_n^{\mathbf{H}}\right]^{\mathbf{H}}$

Update $\mathbf{D}^H\theta$:

$$\mathbf{D}(\mathbf{D}^{\mathbf{H}}\theta) = \mathbf{D}(\zeta - \tilde{\mathbf{c}}/\rho - \tilde{\mathbf{A}}^{\mathbf{H}}\Psi)$$

$$\tilde{\mathbf{A}}^{\mathbf{H}}\Psi = \begin{bmatrix}\mathbf{C}(\mathbf{a}_1)^{\mathbf{H}} \\ \vdots \\ \mathbf{C}(\mathbf{a}_n)^{\mathbf{H}}\end{bmatrix}\Psi = \begin{bmatrix}\mathbf{F}(\hat{\mathbf{a}}_1 \odot \hat{\Psi}^*) \\ \vdots \\ \mathbf{F}(\hat{\mathbf{a}}_n \odot \hat{\Psi}^*)\end{bmatrix}$$

Update ζ:

$$\zeta = \text{sign}(\mathbf{t}) \odot \min(|\mathbf{t}|, \lambda)$$

where $\mathbf{t} = \tilde{\mathbf{A}}^{\mathbf{H}}\Psi + \mathbf{D}^{\mathbf{H}}\theta + \tilde{\mathbf{c}}/\rho$

FFTLasso

FFTLasso

$$\min_{\Psi,\theta} \ \frac{1}{4}\|\Psi\|_2^2 + \Psi^{\mathbf{H}}\mathbf{b} \quad \text{s.t} \ \left\|\tilde{\mathbf{A}}^{\mathbf{H}}\Psi + \mathbf{D}^{\mathbf{H}}\theta\right\|_\infty \leq \lambda$$

Update $\tilde{\mathbf{c}}$:

$$\tilde{\mathbf{c}} \leftarrow \tilde{\mathbf{c}} + \rho(\cancel{\tilde{\mathbf{A}}^{\mathbf{H}}\Psi} + \mathbf{D}^{\mathbf{H}}\theta - \zeta)$$

Nesterov's
Update step:

$$\boxed{\begin{aligned} \tilde{\mathbf{c}}_{k+1} &\leftarrow \tilde{\mathbf{y}}_k + \rho(\tilde{\mathbf{A}}^{\mathbf{H}}\Psi + \mathbf{D}^{\mathbf{H}}\theta - \zeta) \\ \tilde{\mathbf{y}}_{k+1} &\leftarrow (1+q)\tilde{\mathbf{c}}_{k+1} - q\tilde{\mathbf{c}}_k \end{aligned}}$$

where $q = \frac{\sqrt{Q}-1}{\sqrt{Q}+1}$

Q is the problem's condition number which is set as a hyper parameter

Computational Analysis

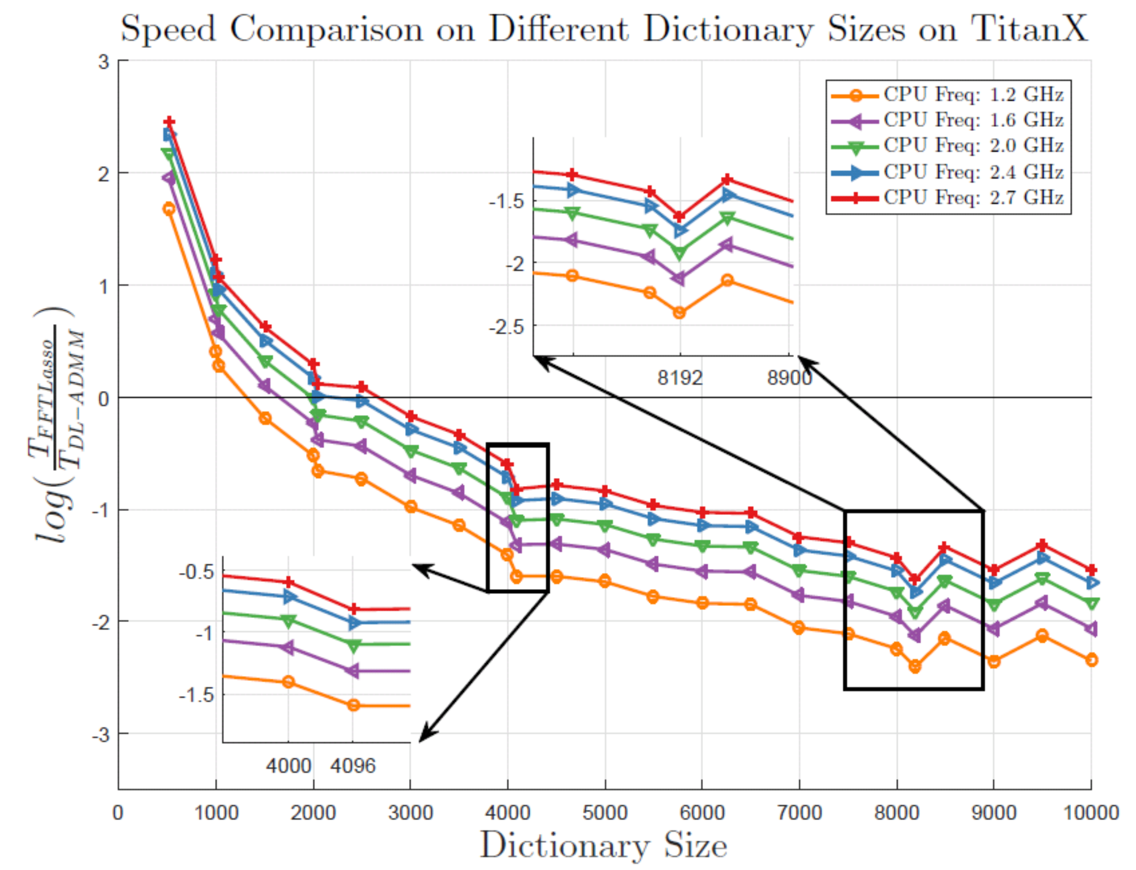
DL-ADMM complexity v. FFTLasso's complexity

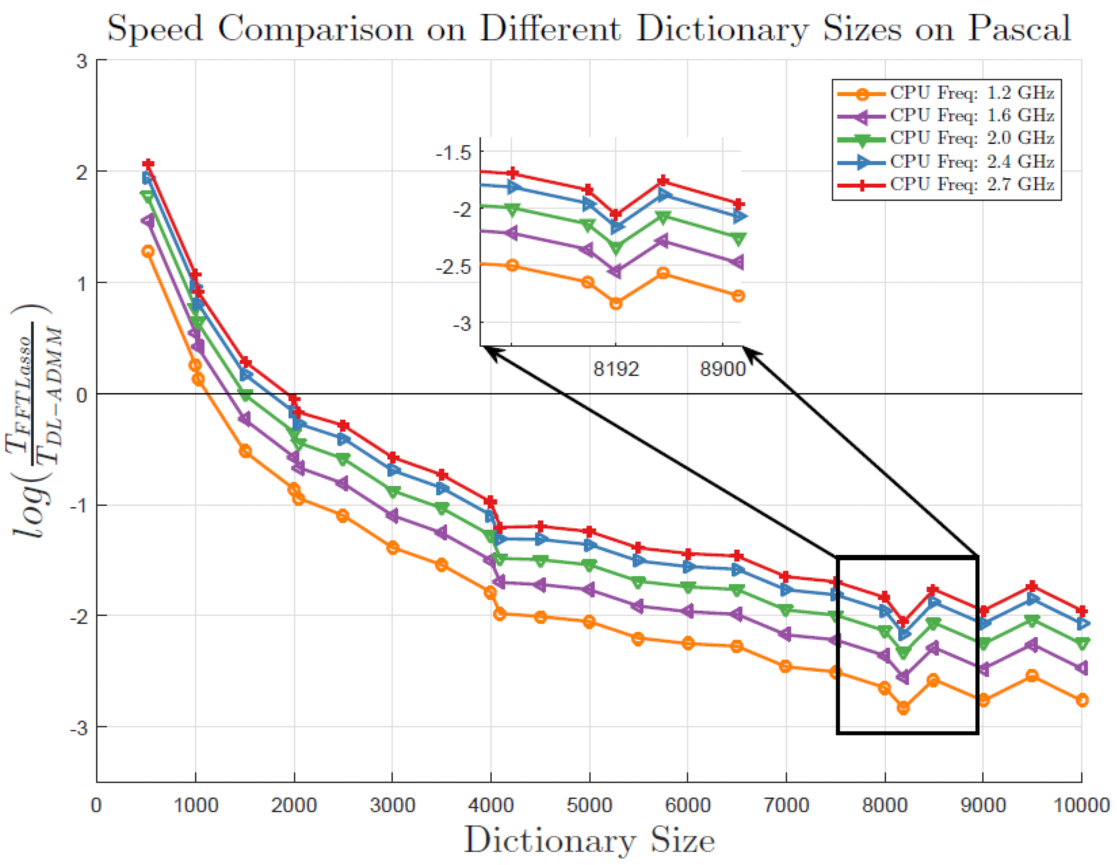
$$\mathcal{O}(m^3) + \mathcal{O}(mn)$$

$$\mathcal{O}(mn \log m) + \mathcal{O}(mn)$$

Dictionaries with $m^2 \gg n \log m$ are best suited for FFTLasso, where square matrices naturally satisfy the inequality

Synthetic Experiments

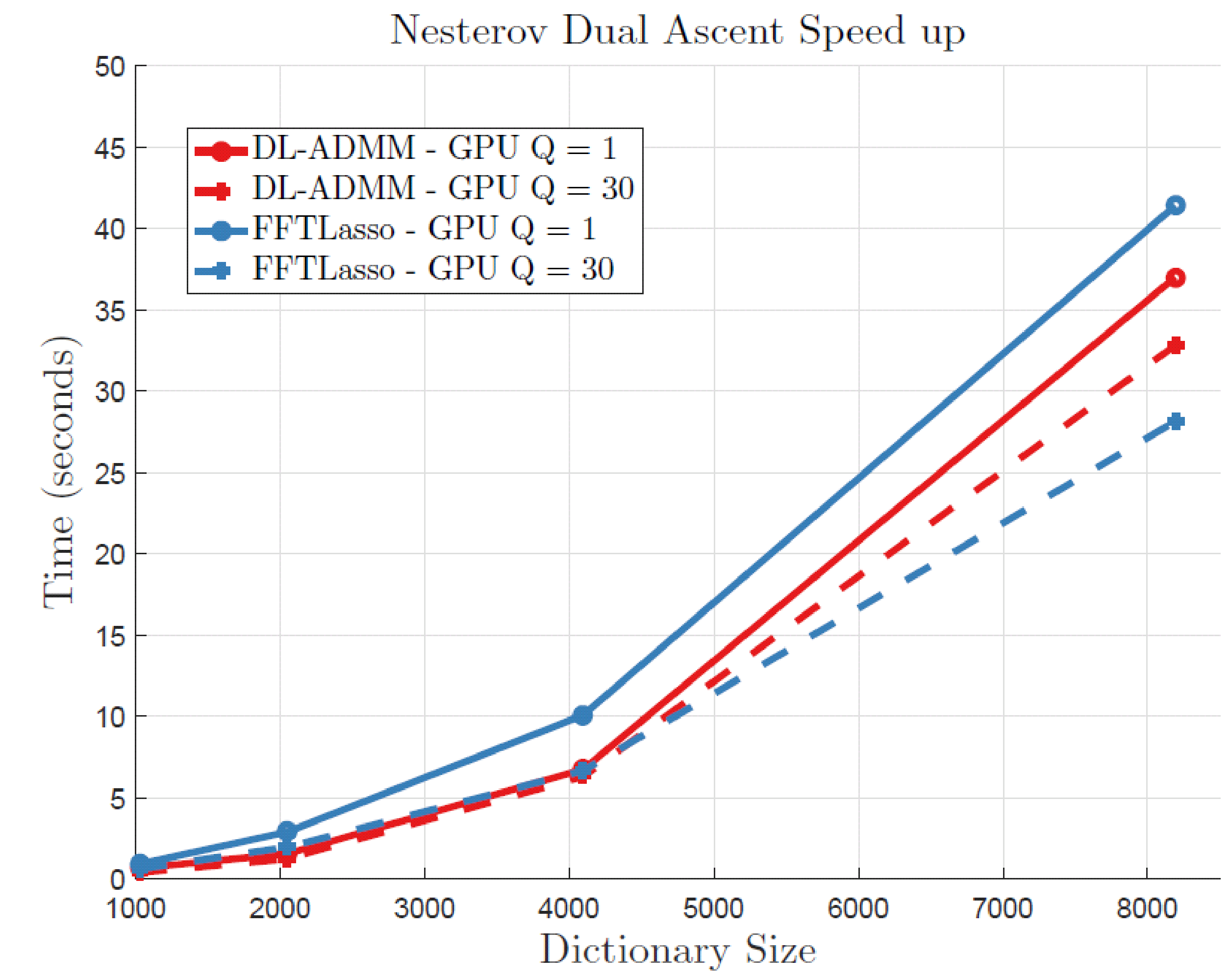




Comparing DL-ADMM on multi-cores with varying frequency against FFTLasso-GPU

Synthetic Experiments

Pascal Titan X:



Synthetic Experiments

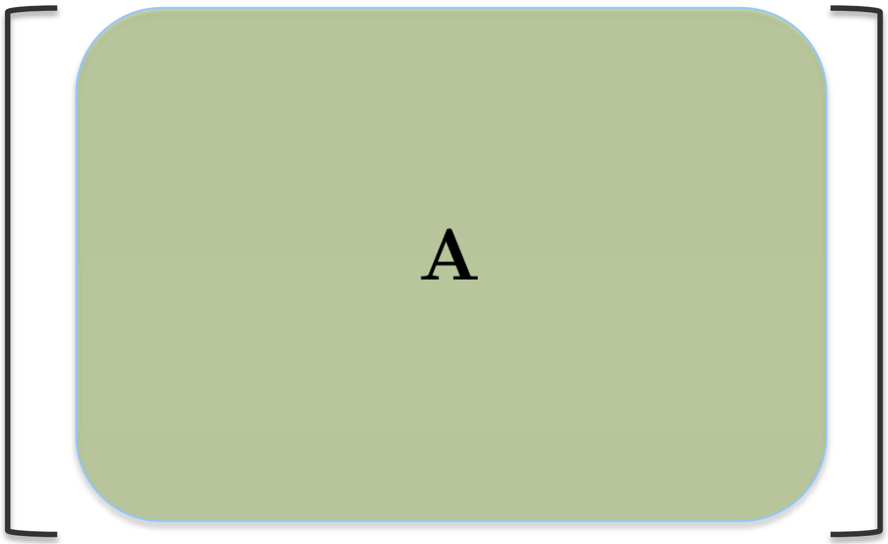
Quadro P6000:

Table 1: GPU time comparison

Dimension	$T_{DLADMM+Nes}/T_{FFTLasso}$
1024	0.043
2048	0.241
4096	1.214
8192	9.792
16384	6.159

GPU Distributed FFTLasso

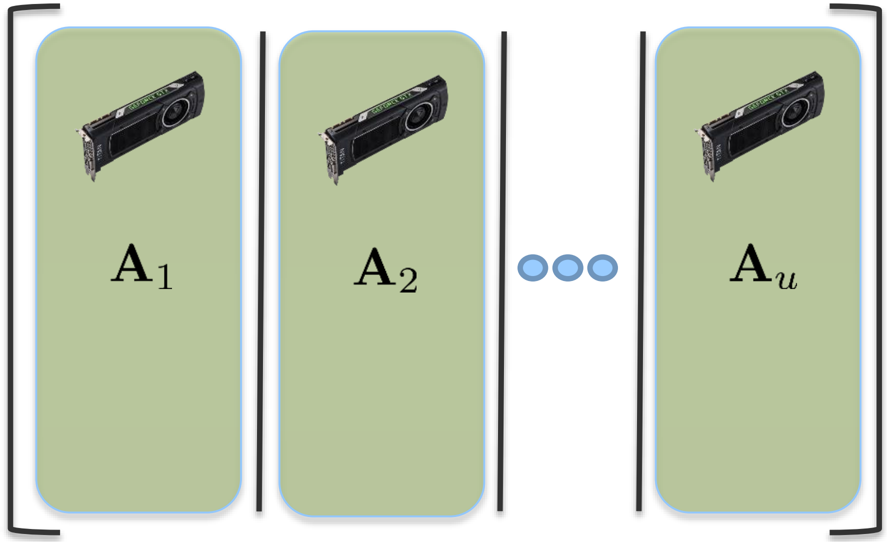
- What if the dictionary is so large that it cannot fit in one GPU?



GPU Distributed FFTLasso

- What if the dictionary is so large that it cannot fit in one GPU?

Easy! <u>Break the dictionary into arbitrary number of vertical pieces. Put each piece on a GPU!</u>

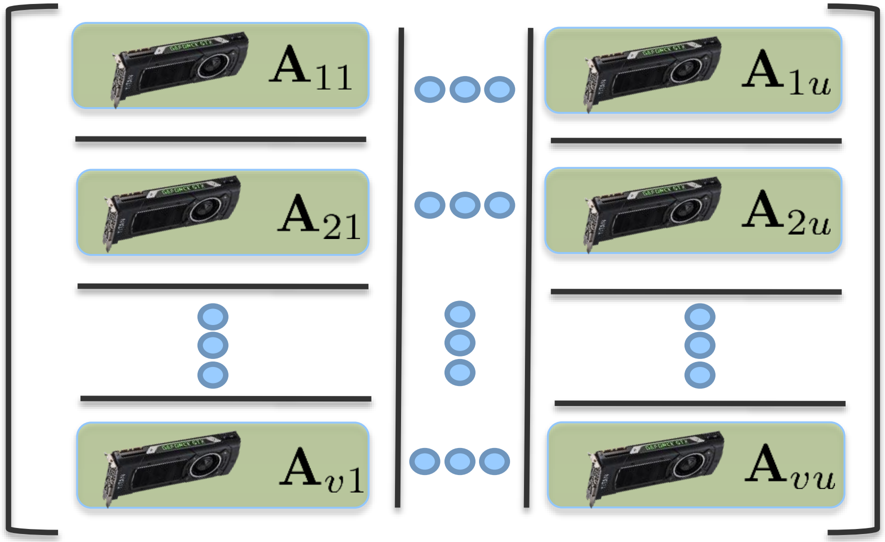


* The necessary overhead is m ; where u is the number of GPUs (for every vertical split).

GPU Distributed FFTLasso

- Can we do better?

Yes! Break it into horizontal pieces too.
Break them into (even and odd pieces) to make best use of FFT properties

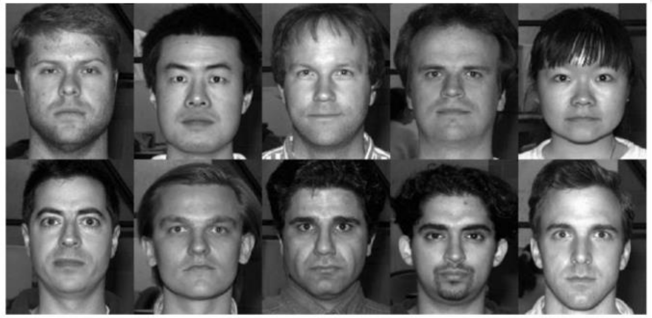


* The necessary overhead is $\frac{m}{v}$

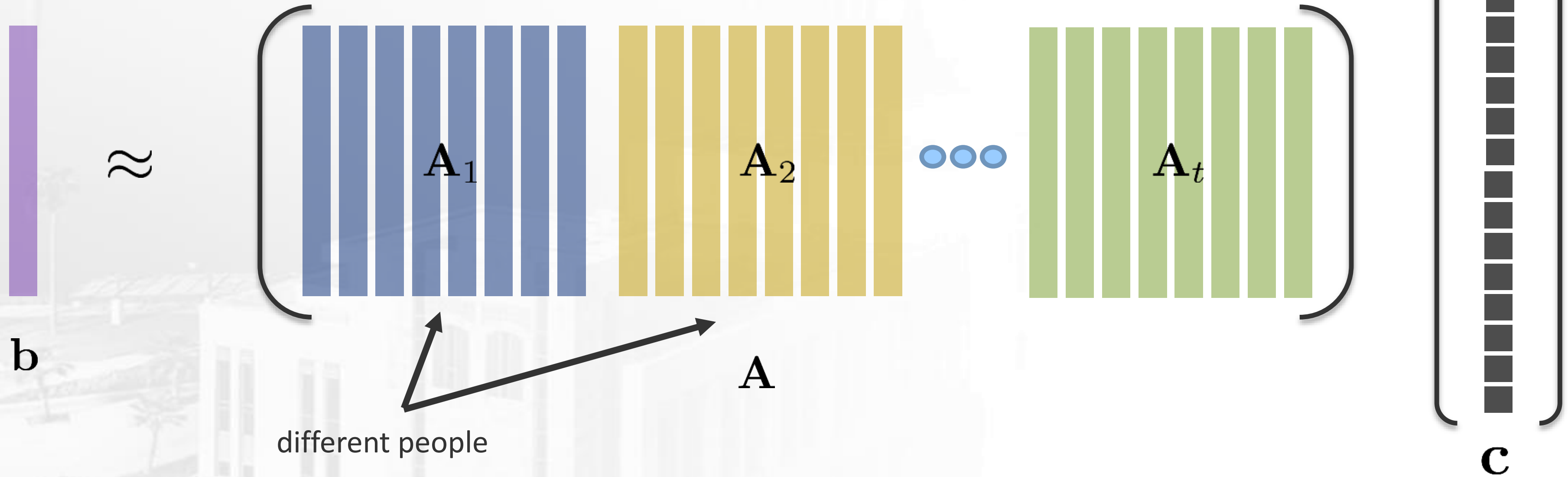
$$\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 \text{ where } \mathbf{v}_1 = \begin{bmatrix} v0 & 0 & v2 & \dots \end{bmatrix}^{\mathbf{H}} \text{ and where } \mathbf{v}_2 = \begin{bmatrix} 0 & v1 & 0 & v3 & \dots \end{bmatrix}^{\mathbf{H}}.$$

$$\tilde{\mathbf{v}} = \begin{bmatrix} \tilde{\mathbf{v}}_e^{\mathbf{H}} & \tilde{\mathbf{v}}_e^{\mathbf{H}} \end{bmatrix}^{\mathbf{H}} + \begin{bmatrix} \tilde{\mathbf{v}}_o^{\mathbf{H}} & \tilde{\mathbf{v}}_o^{\mathbf{H}} \end{bmatrix}^{\mathbf{H}} \odot \mathbf{p}; \quad \mathbf{p}(i,k) = \exp \frac{-j2\pi ki}{m} \quad \forall i, k$$

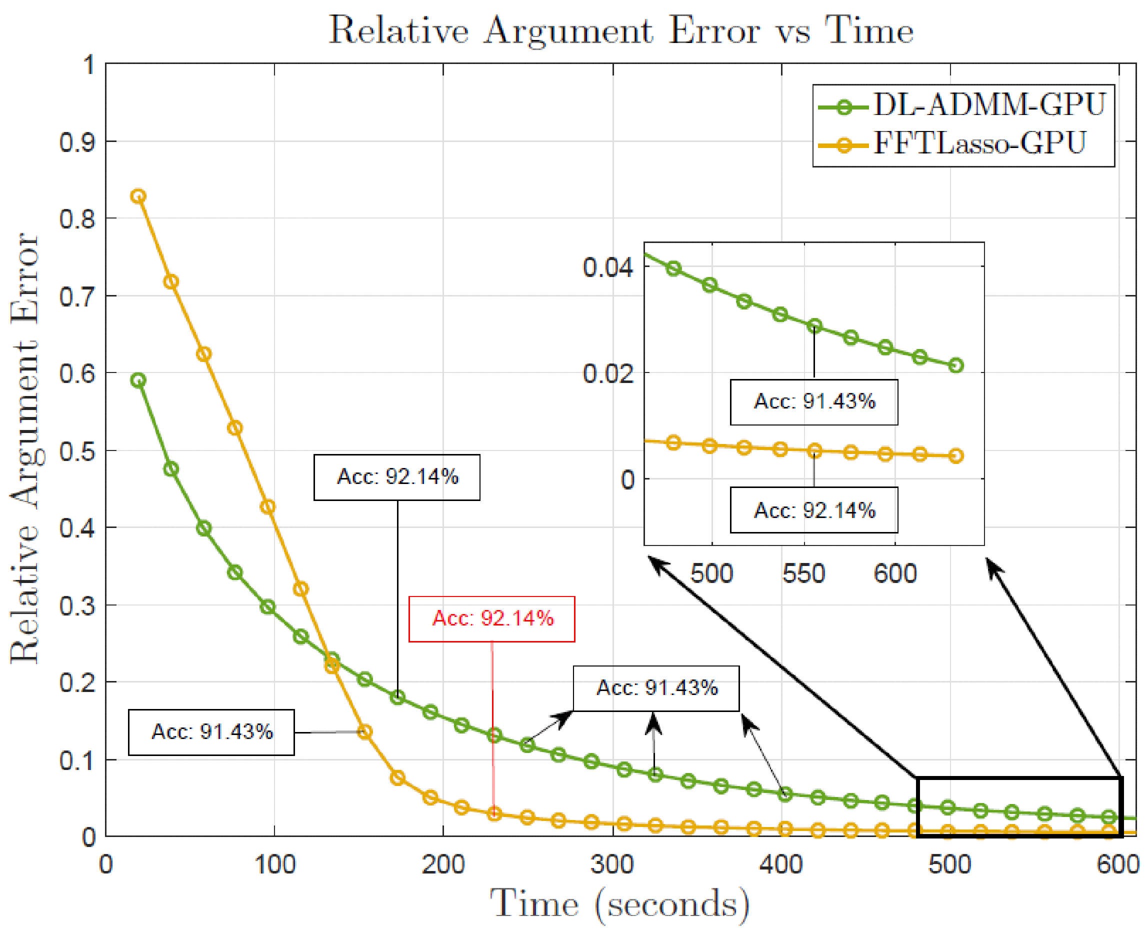
Face Recognition



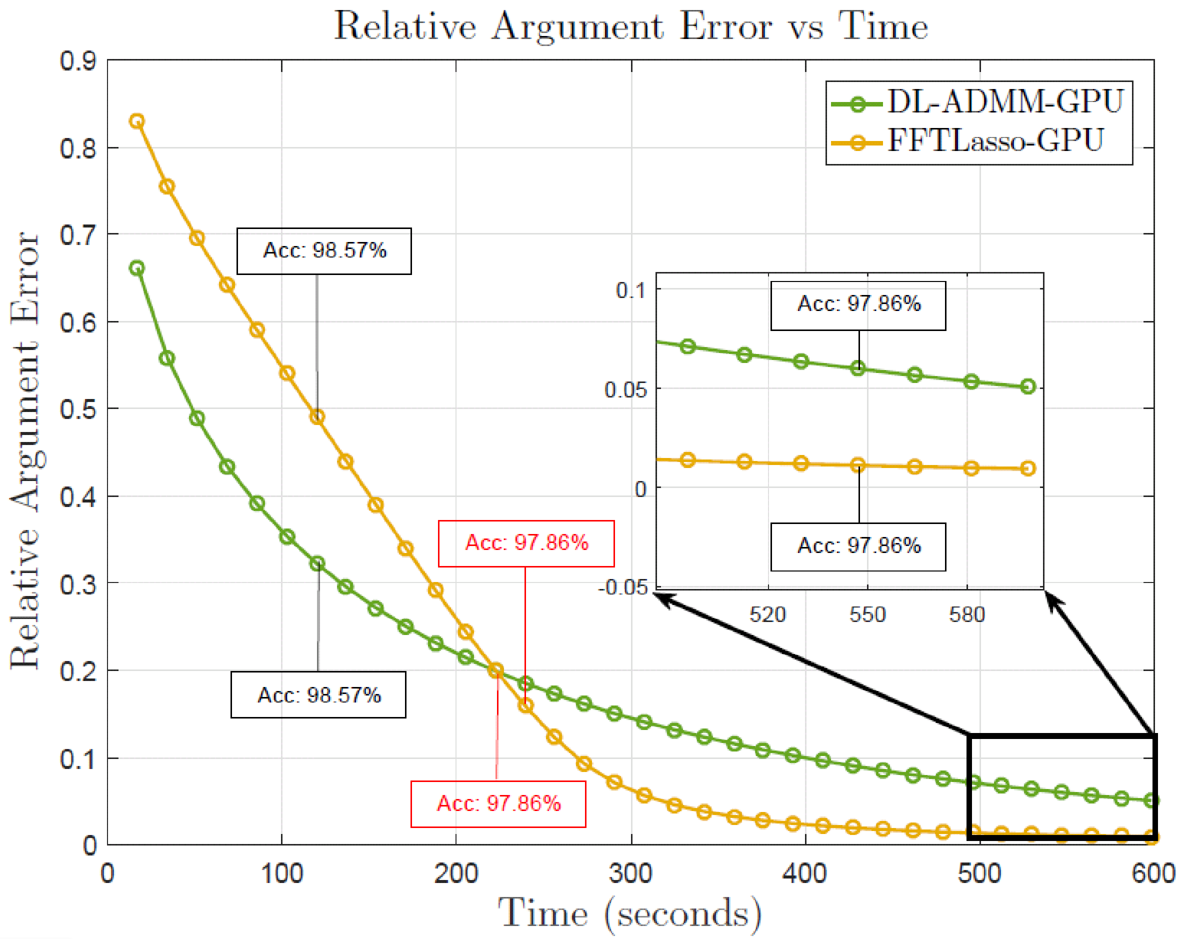
$$\min_{\mathbf{c}} \ \|\mathbf{A}\mathbf{c} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{c}\|_1$$



Face Recognition Experiments



$m = 2^{12}, n = 10^4$



$m = 2^{13}, n = 10^4$

Future Directions

- FFTLasso is easily distributed and also memory efficient (something to talk about at the poster ☺)

- FFTLasso can handle linearly constrained problems (constrained lasso?)

- Other non-smooth regularizers may be used?

Code and Special Thanks!

Congli Wang

Humam Alwassel

Modar Alfadly

NVIDIA CUDA

TensorFlow™

MATLAB

NumPy

Come see us at Poster: #10

https://github.com/adelbibi/FFTLasso