From processing to learning on graphs

Patrick Pérez

Maths and Images in Paris

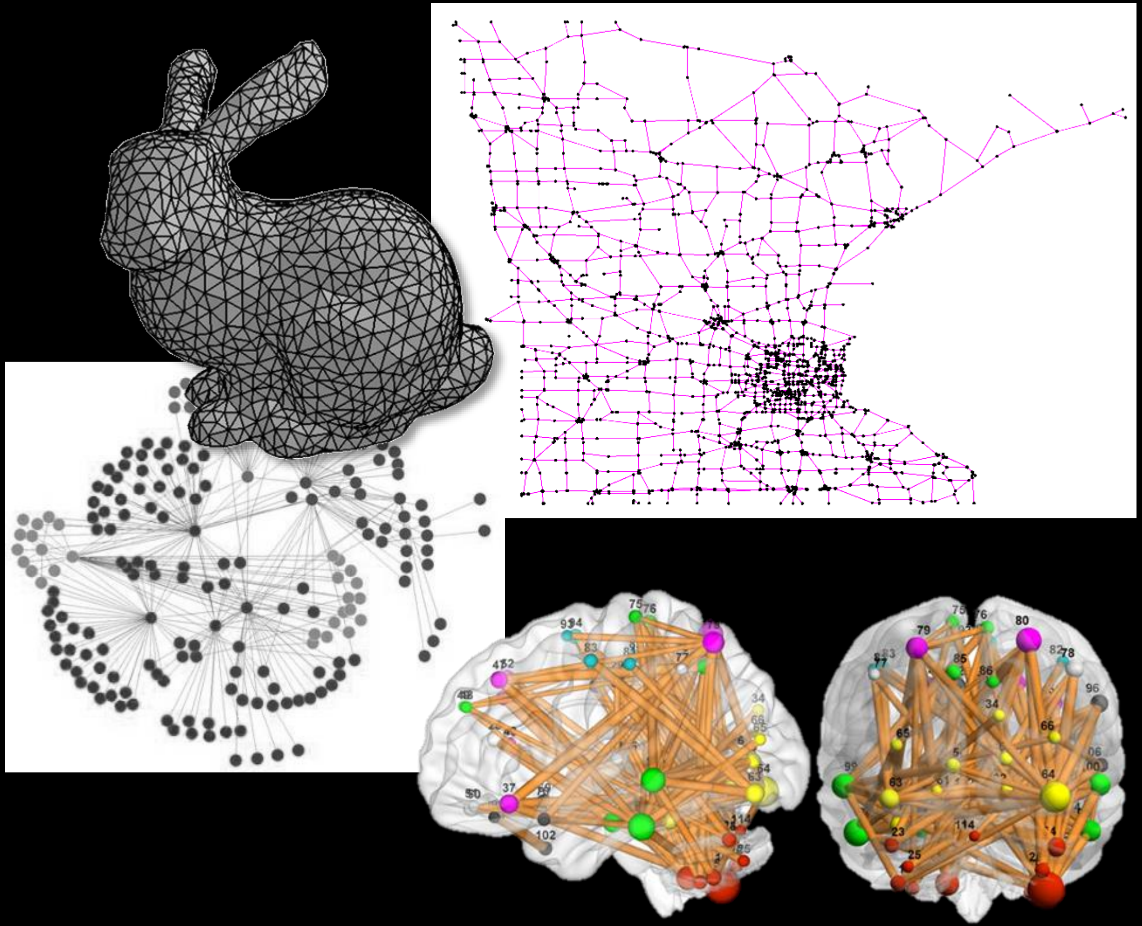
IHP, 2 March 2017

technicolor

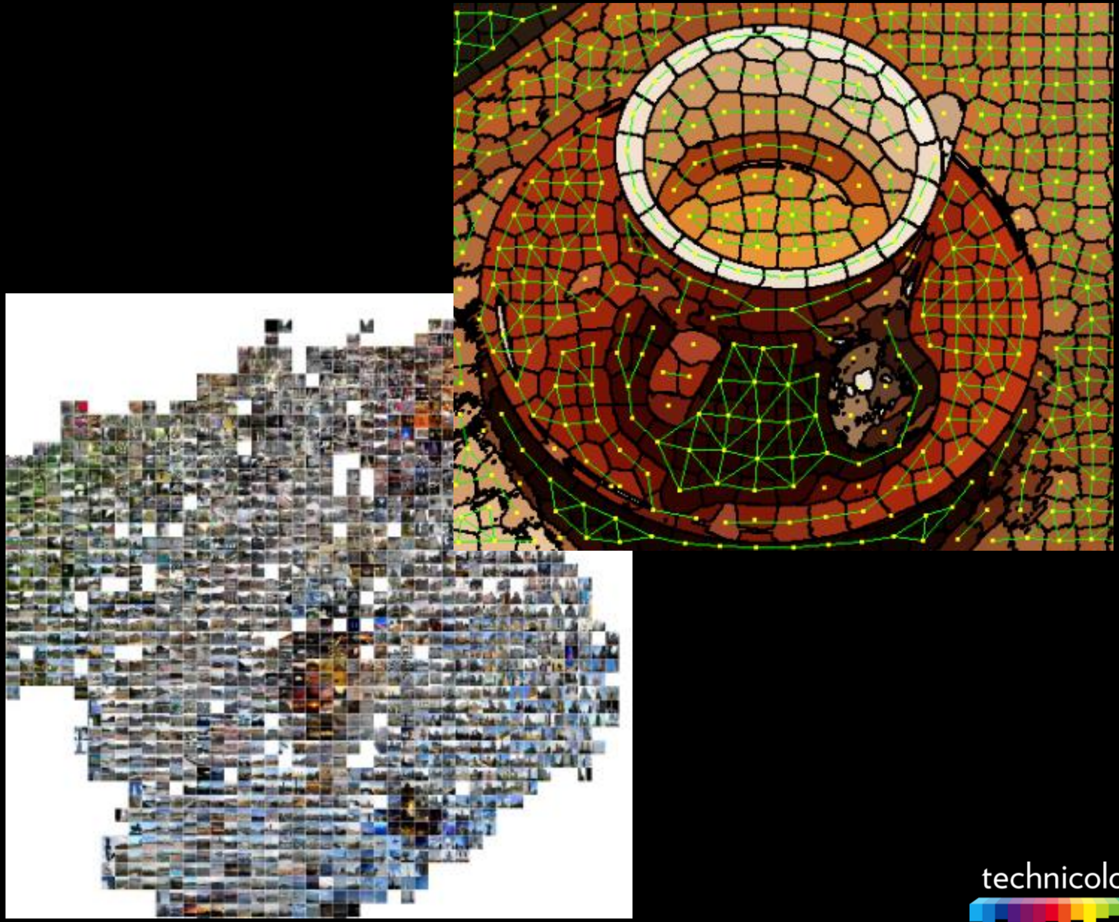
FEEL THE WONDER

Signals on graphs

- Natural graph: mesh, network, etc., related to a “real” structure, various signals can live on it



- Instrumental graph: derived from a collection or a signal, captures its structure, other signals leverage it



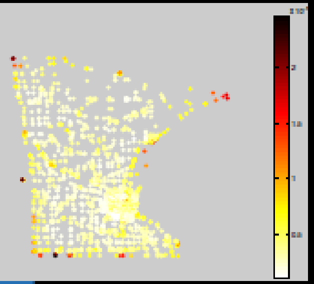
Playing with graph signals

Coding	Compress	Sample	Reconstruct
Processing	Transform	Enhance	Edit
Learning	Cluster	Label	Infer

technicolor
FEEL THE WONDER

Playing with graph signals

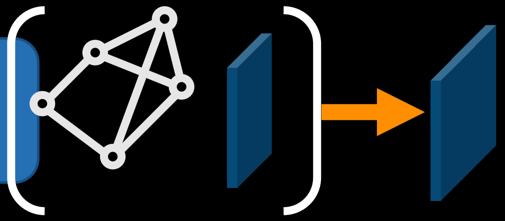
Puy 2016-2017



Coding	Compress	Sample	Reconstruct
Processing	Transform	Enhance	Edit
Learning	Cluster	Label	Infer

Playing with graph signals

Coding	Compress	Sample	Reconstruct
Processing	Transform	Enhance	Edit
Learning	Cluster	Label	Infer

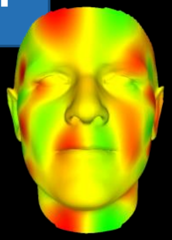


Puy 2017

technicolor
FEEL THE WONDER

Playing with graph signals

Coding	Compress	Sample	Reconstruct
Processing	Transform	Enhance	Edit
Learning	Cluster	Label	Infer



Garrido 2016

Undirected weighted graph

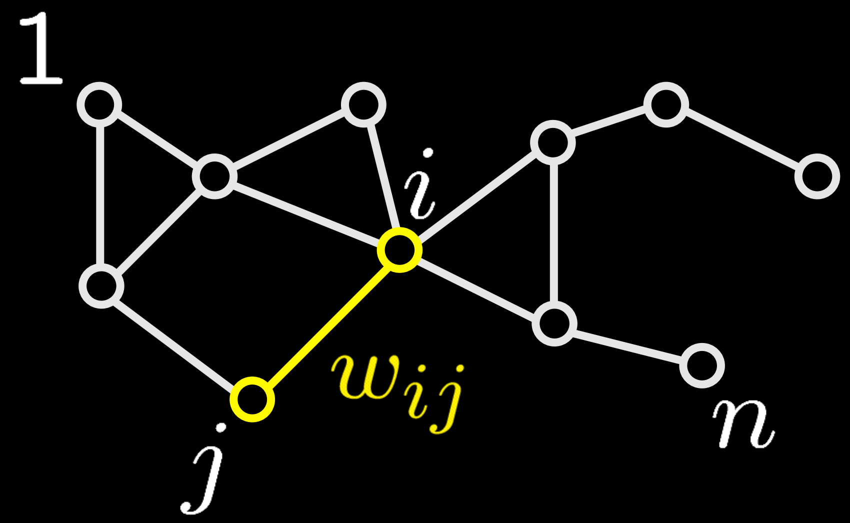
$$\boxed{\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathsf{W}\}}$$

$$\mathcal{V} = (1, n)$$

$$\mathcal{E} \in \mathcal{V} \times \mathcal{V}$$

$$\mathsf{W} = [w_{ij}] \in \mathbb{R}_{+}^{n \times n}, \; sym.$$

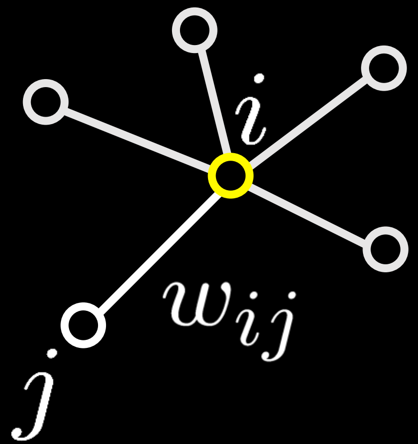
$$w_{ij} > 0 \iff (i, j) \in \mathcal{E}$$



Graph Laplacian(s)

Vertex degree and degree matrix

$$d_i = \sum_{j=1}^{n} w_{ij}, \quad \mathsf{D} = \mathsf{diag}(d_1 \cdots d_n)$$



Symmetric p.s.d. Laplacians

- Combinatorial Laplacian

$$\boxed{\mathsf{L} = \mathsf{D} - \mathsf{W}}$$

- Normalized Laplacian

$$\mathsf{L_{norm}} = \mathsf{Id} - \mathsf{D}^{-1/2}\mathsf{W}\mathsf{D}^{-1/2}$$

$$\mathsf{L}\mathbf{1} = \mathbf{0}$$

$$\mathsf{L_{norm}}\mathsf{D}^{1/2}\mathbf{1} = \mathbf{0}$$

technicolor
FEEL THE WONDER

Graph signal and smoothness

Signals / functions on graph

- Scalar $\boldsymbol{x} \in \mathbb{R}^n, \quad i \mapsto f(i) = x_i$
- Multi-dim. $\mathsf{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_m] \in \mathbb{R}^{n \times m}, \quad i \mapsto \boldsymbol{f}(i) = (x_{k,i})_{k=1}^m$

Graph smoothness

- Scalar

$$\frac{1}{2}\sum_{i,j} w_{ij}(x_i - x_j)^2 = \boldsymbol{x}^\top \mathsf{L} \boldsymbol{x}$$

- Multi-dimensional

$$\frac{1}{2}\sum_{i,j} w_{ij}\|\boldsymbol{f}(i) - \boldsymbol{f}(j)\|^2 = \mathrm{trace}(\mathsf{X}^\top \mathsf{L} \mathsf{X})$$

technicolor
FEEL THE WONDER

Spectral graph analysis

Laplacian diagonalization and graph harmonics of increasing "frequencies"

$$\boxed{\mathsf{L} = \mathsf{U}\Lambda\mathsf{U}^\top}$$

$$\lambda_1 = 0 \leqslant \lambda_2 \cdots \leqslant \lambda_n \leqslant 2\max\{d_i\}$$
$$\Lambda = \mathrm{diag}(\lambda_1 \cdots \lambda_n)$$
$$\mathsf{U} = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_n] \text{ orthogonal}$$
$$\boldsymbol{u}_\ell^\top \mathsf{L} \boldsymbol{u}_\ell = \lambda_\ell$$

Graph Fourier transform and its inverse

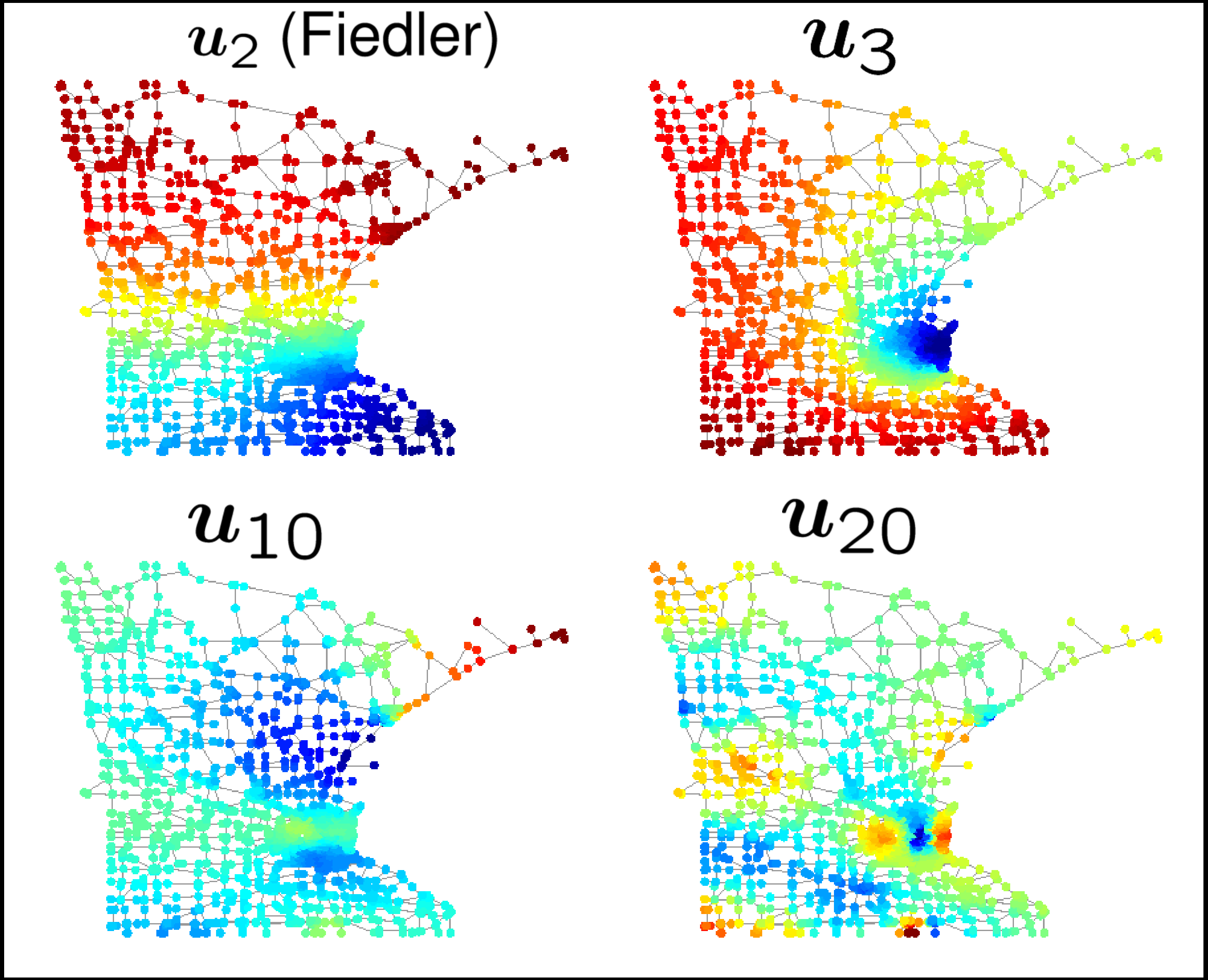
$$\boxed{\hat{\boldsymbol{x}} = \mathsf{U}^\top \boldsymbol{x}} \qquad \boldsymbol{x} = \mathsf{U}\hat{\boldsymbol{x}}$$

Smooth (*k*-bandlimited) signals

$$\boldsymbol{x} \in \mathrm{span}(\mathsf{U}_k), \; \mathsf{U}_k = [\boldsymbol{u}_1 \cdots \boldsymbol{u}_k]$$

technicolor
FEEL THE WONDER

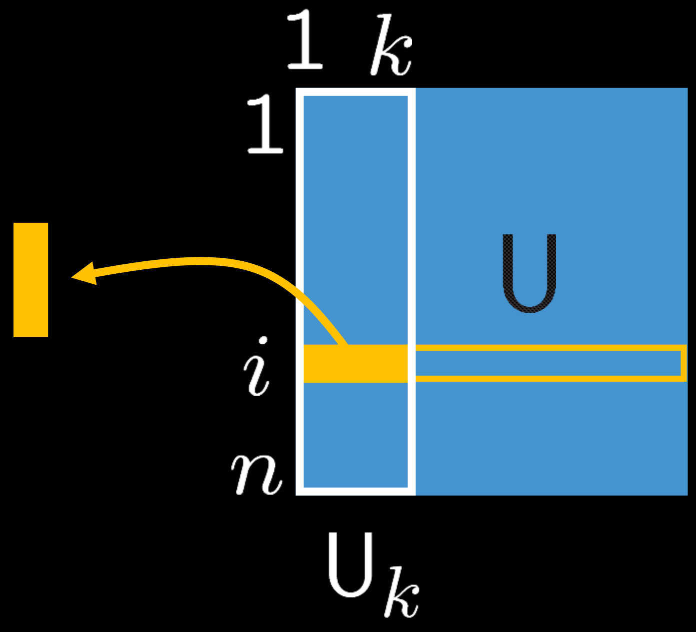
Spectral graph analysis



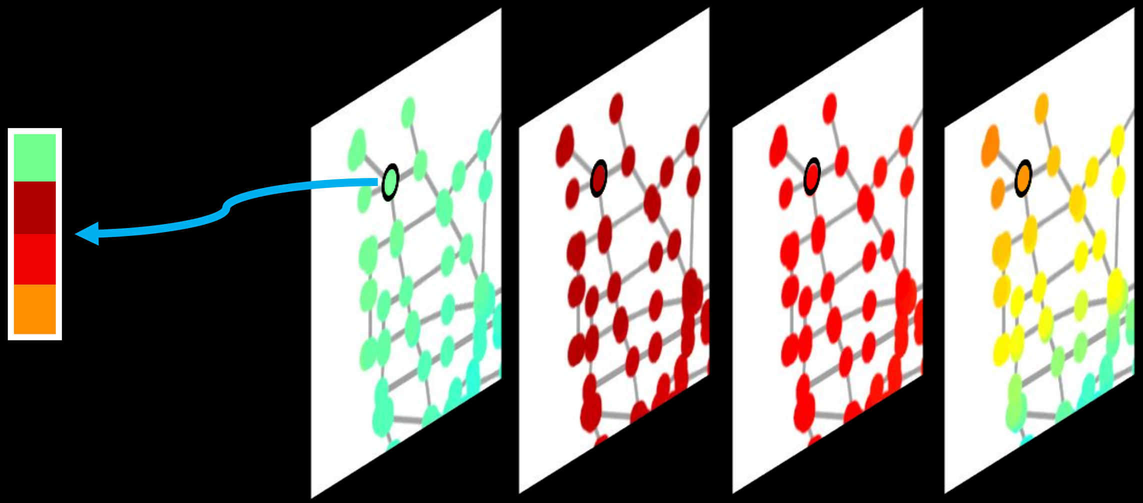
Spectral vertex embedding

Rows of truncated Fourier basis

$$\boldsymbol{b}_i = \mathsf{U}_k^\top \boldsymbol{\delta}_i \in \mathbb{R}^k, \; i = 1 \cdots n$$



⇒ *k*-dim embedding of vertices



Clustered with *k*-means in *spectral clustering*

Linear filters and convolutions

Filtering in the spectral domain

- ▶ With filter Fourier transform

$$\hat{\boldsymbol{h}} \in \mathbb{R}^n$$

$$\boldsymbol{h} \star \boldsymbol{x} = \mathsf{U}\mathsf{diag}(\hat{\boldsymbol{h}})\mathsf{U}^\top \boldsymbol{x}$$

- ▶ Through frequency filtering

$$\hat{g} : \mathbb{R}_+ \to \mathbb{R}_+$$

$$g(\boldsymbol{x}) = \mathsf{U}\hat{g}(\Lambda)\mathsf{U}^\top \boldsymbol{x}$$

Issues
- locality on graph
- computational complexity

Polynomial filtering: from spectral to vertex domain

- ▶ Controlled locality and complexity

$$\hat{g}(\lambda) = \sum_{r=1}^{d} \alpha_r \lambda^r$$

$$g(\boldsymbol{x}) = \mathsf{U}\hat{g}(\Lambda)\mathsf{U}^\top = \sum_{r=1}^{d} \alpha_r \mathsf{L}^r \boldsymbol{x}$$

Sampling graph signals

Random sampling

- Define vertex sampling distribution
- Draw signal samples accordingly

$$\boldsymbol{p} \in [0,1]^n, \; \|\boldsymbol{p}\|_1 = 1$$

$$\omega_j \sim \boldsymbol{p}, \; j = 1 \cdots m$$

$$\boldsymbol{y} = \mathrm{M}\boldsymbol{x} = (x_{\omega_j})_{j=1}^m$$

Problems

- Reconstruction of *smooth* signals
- Performance as function of *m*
- Best sampling distribution

[Puy *et al.* 2016]

Reconstructing smooth signals from samples

Smooth interpolation / approximation (noisy measures)

$$\arg\min_{z \in \mathbb{R}^n} z^\top \mathsf{L} z \text{ sb.t. } \mathsf{M}z = y \qquad \arg\min_{z \in \mathbb{R}^n} \|y - \mathsf{M}z\|^2 + \gamma z^\top \mathsf{L} z$$

k-bandlimited approximation: exact or approximate

$$\arg\min_{z \in \mathrm{span}(\mathsf{U}_k)} \|y - \mathsf{M}z\|^2$$

$$\arg\min_{z \in \mathbb{R}^n} \|y - \mathsf{M}z\|^2 + \gamma z^\top \hat{g}(\mathsf{L}) z$$

with $\hat{g}$ a highpass polynom.filter

$\hat{g}$ non-decreasing

$\hat{g}(\lambda_k)$ small, $\hat{g}(\lambda_{k+1}) > 0$

[Puy *et al.* 2016]

technicolor
FEEL THE WONDER

Reconstruction quality (1)

$$z^* \in \underset{z \in \text{span}(\mathsf{U}_k)}{\arg\min} \left\| \mathsf{P}^{-1/2}(y - \mathsf{M}z) \right\|^2$$

$$\mathsf{P} = \text{diag}(p_{\omega_j})$$

Assuming RIP*

- Noisy measurements: $y = \mathsf{M}x + n$

Given $\varepsilon, \delta \in (0,1)$, with proba. at least $1-\varepsilon$,
$\forall x \in \text{span}(\mathsf{U}_k)$, n:

$$\|z^* - x\| \leqslant \frac{2}{\sqrt{m(1-\delta)}} \left\| \mathsf{P}^{-1/2} n \right\|$$

- Noiseless measurements: exact recovery

* *m* large enough, for now

technicolor
FEEL THE WONDER

Reconstruction quality (2)

$$z^* = \underset{z \in \mathbb{R}^n}{\arg\min} \|\mathsf{P}^{-1/2}(y - \mathsf{M}z)\|_2^2 + \gamma z^\top \hat{g}(\mathsf{L}) z$$

Assuming RIP*

Given $\varepsilon, \delta \in (0,1)$, with proba. at least $1-\varepsilon$,

$\forall x \in \mathrm{span}(\mathsf{U}_k),\ n$:

$$\|\mathsf{U}_k^\top \mathsf{U}_k z^* - x\| \leqslant \frac{1}{\sqrt{m(1-\delta)}} \left(A \left\| \mathsf{P}^{-1/2} n \right\| + B\|x\| \right).$$

$$\|\mathsf{U}_k^\top \mathsf{U}_k z^* - z^*\| \leqslant C \left\| \mathsf{P}^{-1/2} n \right\| + D\|x\|$$

* *m* large enough, for now

technicolor

FEEL THE WONDER

Optimizing sampling

Some vertices are more important

- Norm of spectral embedding: max. energy fraction on vertex from *k*-bandlimited signal

$$\|\boldsymbol{b}_i\| = \|\mathsf{U}_k^\top \boldsymbol{\delta}_i\| = \frac{\|\mathsf{U}_k^\top \boldsymbol{\delta}_i\|}{\|\mathsf{U}^\top \boldsymbol{\delta}_i\|} \leqslant 1 \qquad \|\mathsf{U}_k^\top \boldsymbol{\delta}_i\| = \max_{\boldsymbol{\eta}\in\mathbb{R}^k} \frac{\|\boldsymbol{\eta}^\top \mathsf{U}_k^\top \boldsymbol{\delta}_i\|}{\|\boldsymbol{\eta}\|}$$

$\|\boldsymbol{b}_i\| \approx 1$ Exists a *k*-bandlimited signal concentrated on this node; should be sampled

$\|\boldsymbol{b}_i\| \approx 0$ Exists no *k*-bandlimited signal concentrated on this node; can be ignored

- Graph weighted coherence of distribution

$$\nu_{\boldsymbol{p}}^k = \max_{1\leqslant i\leqslant n} \left\{ p_i^{-1/2} \|\boldsymbol{b}_i\| \right\} \geqslant \sqrt{k}$$

should be as small as possible

technicolor

FEEL THE WONDER

Restricted Isometry Property (RIP)

Given $\varepsilon, \delta \in (0,1)$, with proba. at least $1-\varepsilon$,

$$(1-\delta)\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 \leqslant \frac{1}{m}\left\|\mathsf{P}^{-1/2}\mathsf{M}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right\|^2 \leqslant (1+\delta)\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2$$

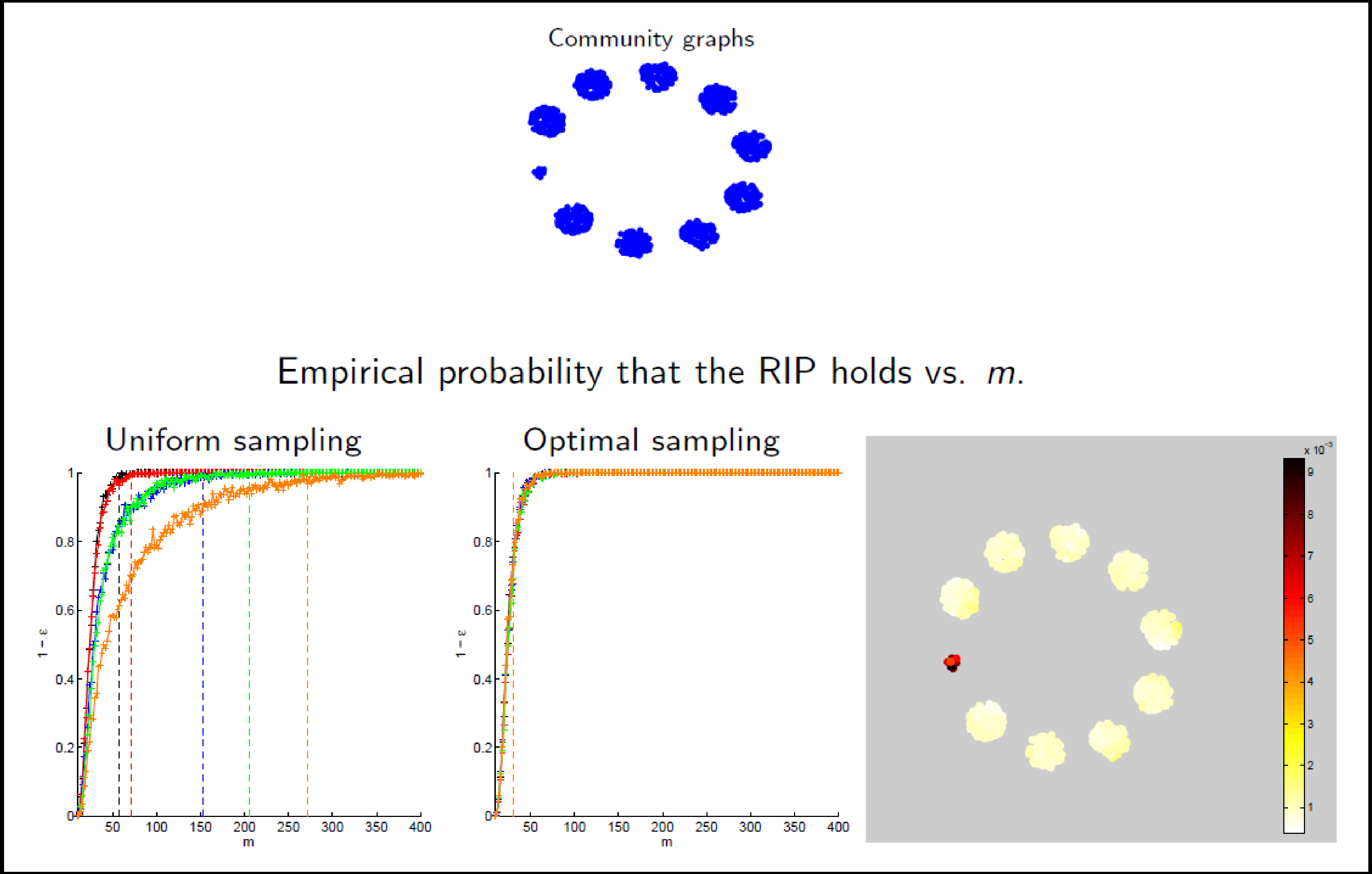
for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \text{span}(\mathsf{U}_k)$ if

$$m \geqslant \frac{3}{\delta^2}(\nu_{\boldsymbol{p}}^k)^2 \ln\left(\frac{2k}{\varepsilon}\right)$$

- $(\nu_{\boldsymbol{p}}^k)^2 \ln(k)$ vertices are enough to sample all *k*-bandlimited signals
- In best case, $k \ln(k)$ suffice
- Once selected, vertices can be used to sample all *k*-bandlimited signals

technicolor
FEEL THE WONDER

Empirical RIP



technicolor
FEEL THE WONDER

Optimal and practical sampling

Optimal sampling distribution

$$p_i^* = k^{-1} \|\mathsf{U}_k^\top \boldsymbol{\delta}_i\|^2 \Rightarrow \boldsymbol{\nu}_{\boldsymbol{p}^*}^k = \sqrt{k}$$

- $k \ln(k)$ measurements suffice, but requires computation of harmonics

Efficient approximation

- Rapid computation of alternative vertex embedding of similar norms

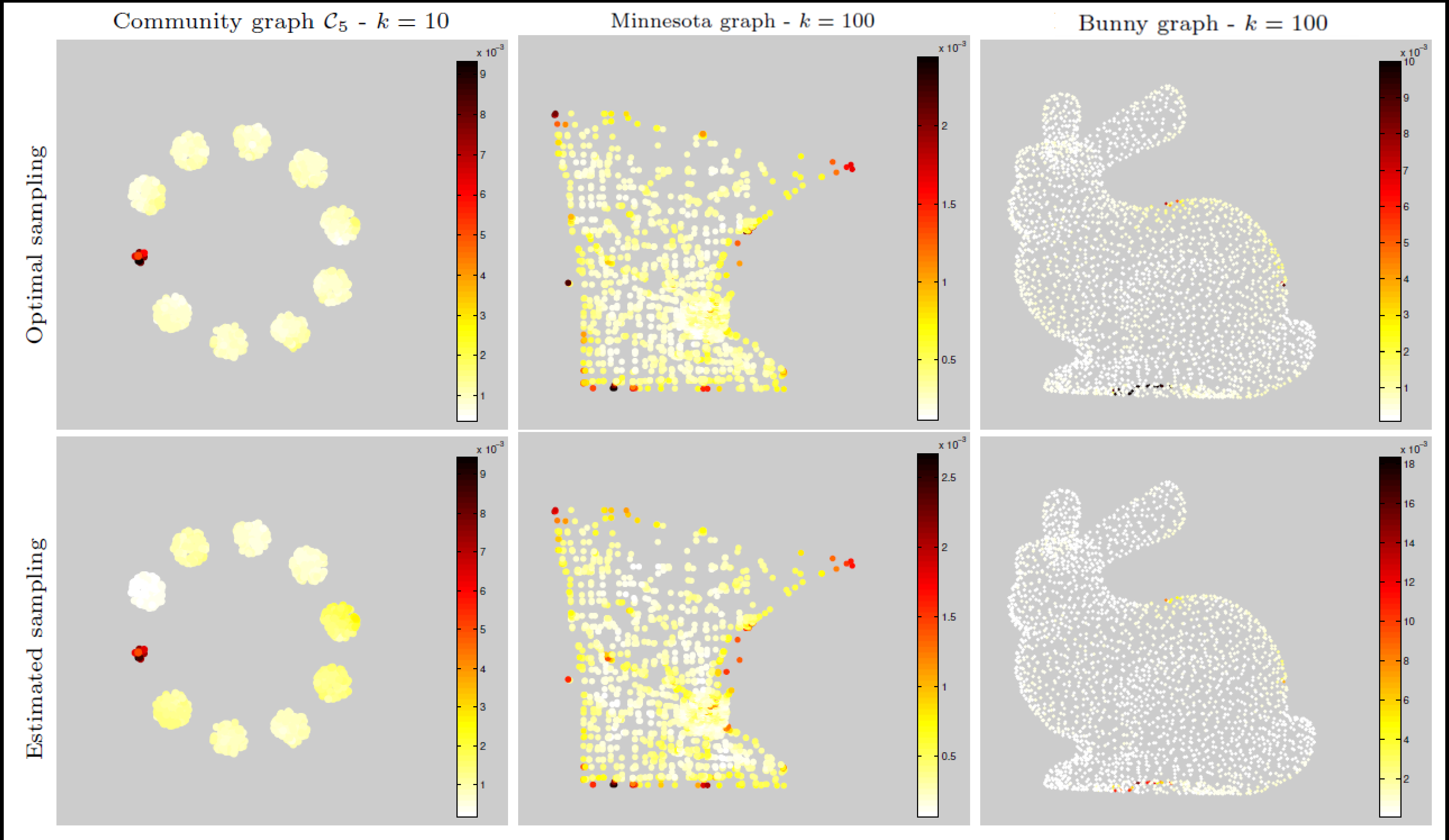
$$\tilde{\boldsymbol{b}}_i = \mathsf{R}_{n \times \ell}^\top \boldsymbol{\delta_i}$$

with columns of R obtained by polynomial filtering of suitable Gaussian signals

- Can serve also for efficient spectral clustering [Tremblay *et al.* 2016]

technicolor
FEEL THE WONDER

Optimal and practical sampling



Extension to group sampling

[Puy and Pérez 2017]
under submission

Given a suitable partition of vertices $\mathcal{V} = \cup_{\ell=1}^{N} \mathcal{V}_\ell$

- Smooth graph signals **almost piece-wise constant on groups**

Random sampling?
Reconstruction?

Interest

- Speed and memory gains (working on reduced signal versions)
- Interactive systems: propose sampled groups for user to annotate

technicolor
FEEL THE WONDER

Extension to group sampling

[Puy and Pérez 2017]
under submission

Given a suitable partition of vertices $\mathcal{V} = \cup_{\ell=1}^{N} \mathcal{V}_\ell$

- Smooth graph signals **almost piece-wise constant on groups**

Random sampling?
Reconstruction?

Interest

- Speed and memory gains (working on reduced signal versions)
- Interactive systems: propose sampled groups for user to annotate

technicolor
FEEL THE WONDER

Group sampling and group coherence

Reasoning at group level

- Group sampling

$$\boldsymbol{p} \in [0,1]^N, \; \|\boldsymbol{p}\|_1 = 1$$

$$\omega_j \sim \boldsymbol{p}, \; j = 1 \cdots s$$

$$\boldsymbol{y} = \mathsf{M}\boldsymbol{x} = (x_i)_{i \in \mathcal{V}_{\omega_j}, j=1\cdots s}$$

$$m = \sum_{j=1}^{s} |\mathcal{V}_{\omega_j}|$$

- Local group coherence: max energy fraction in group from a *k*-bandlimited signal*

$$\max_{\boldsymbol{\eta} \in \mathbb{R}^k} \frac{\|\mathsf{N}^\ell \mathsf{U}_k \boldsymbol{\eta}\|}{\|\boldsymbol{\eta}\|} = \|\mathsf{N}^\ell \mathsf{U}_k\|_2$$

- Group coherence:

$$\nu_{\boldsymbol{p}}^k = \max_{1 \leqslant \ell \leqslant N} \left\{ p_\ell^{-1/2} \|\mathsf{N}^\ell \mathsf{U}_k\|_2 \right\} \geqslant 1$$

* $\mathsf{N}^{(\ell)} \boldsymbol{x} = (x_i)_{i \in \mathcal{V}_\ell}$

technicolor

FEEL THE WONDER

Restricted Isometry Property (RIP)

$\mathsf{P} = \mathrm{diag}\left(p_{\omega_j} \mathrm{Id}_{|\mathcal{V}_{\omega_j}|}\right)$

Given $\varepsilon, \delta \in (0, 1)$, with proba. at least $1 - \varepsilon$

$$(1-\delta)\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 \leqslant \frac{1}{s}\left\|\mathsf{P}^{-1/2}\mathsf{M}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right\|^2 \leqslant (1+\delta)\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2$$

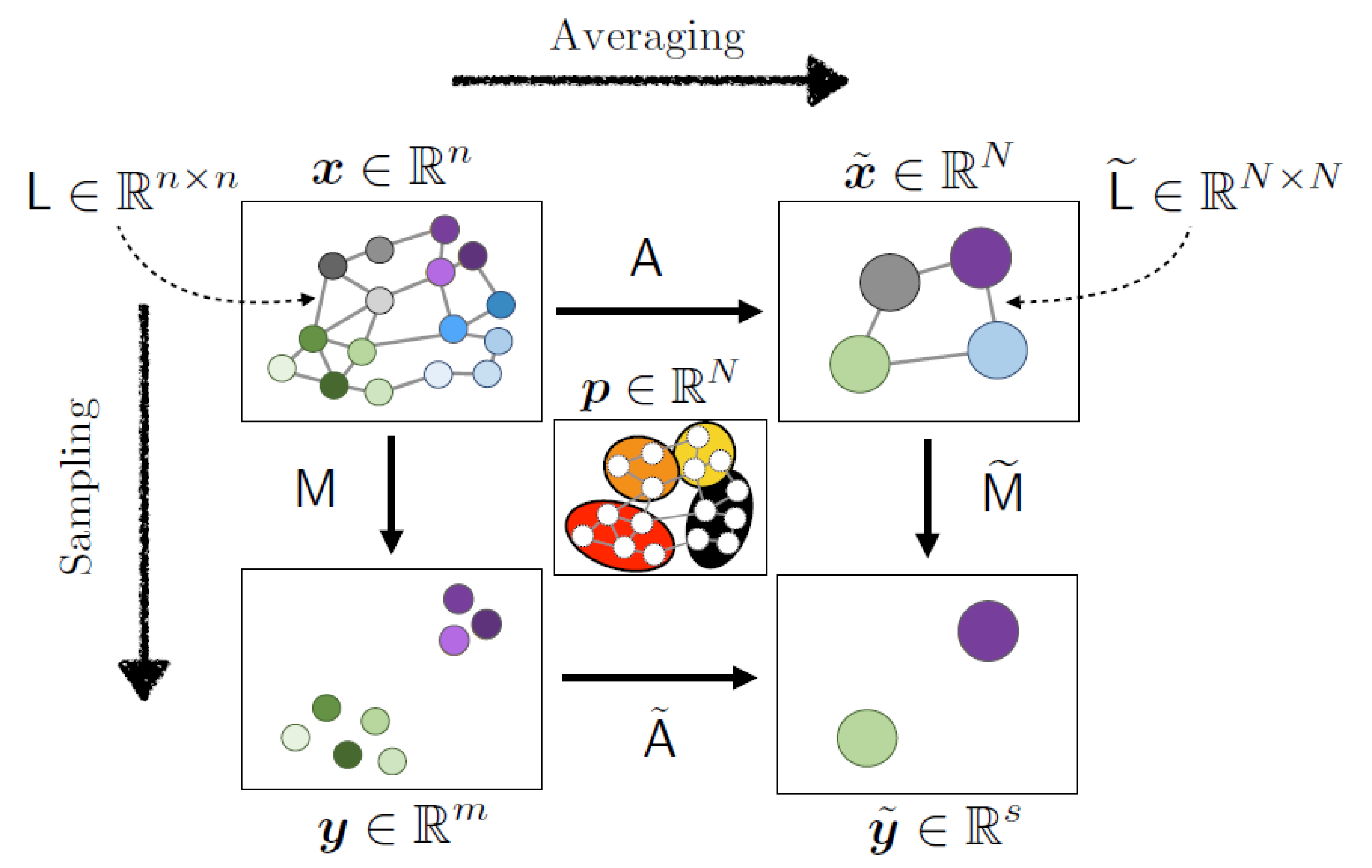
for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathrm{span}(\mathsf{U}_k)$ if

$$s \geqslant \frac{3}{\delta^2}(\nu_{\boldsymbol{p}}^k)^2 \ln\left(\frac{2k}{\varepsilon}\right)$$

- $(\nu_{\boldsymbol{p}}^k)^2 \ln(k)$ groups are enough to sample all *k*-bandlimited signals
- In best case, $\ln(k)$ groups suffice

Smooth piece-wise constant reconstruction

$$\tilde{z}^* = \arg\min_{\tilde{z} \in \mathbb{R}^N} \|\tilde{\mathsf{P}}^{-1/2}(\tilde{\mathsf{A}}y - \tilde{\mathsf{M}}\tilde{z})\|_2^2 + \gamma \tilde{z}^\top \hat{g}(\tilde{\mathsf{L}})\tilde{z}$$



Smooth piece-wise constant reconstruction

$$\tilde{\boldsymbol{z}}^* = \arg\min_{\tilde{\boldsymbol{z}} \in \mathbb{R}^N} \|\tilde{\mathsf{P}}^{-1/2}(\tilde{\mathsf{A}}\boldsymbol{y} - \tilde{\mathsf{M}}\tilde{\boldsymbol{z}})\|_2^2 + \gamma \tilde{\boldsymbol{z}}^\top \hat{g}(\tilde{\mathsf{L}})\tilde{\boldsymbol{z}}$$

Assuming RIP

Given $\varepsilon, \delta \in (0,1)$, with proba. at least $1-\varepsilon$

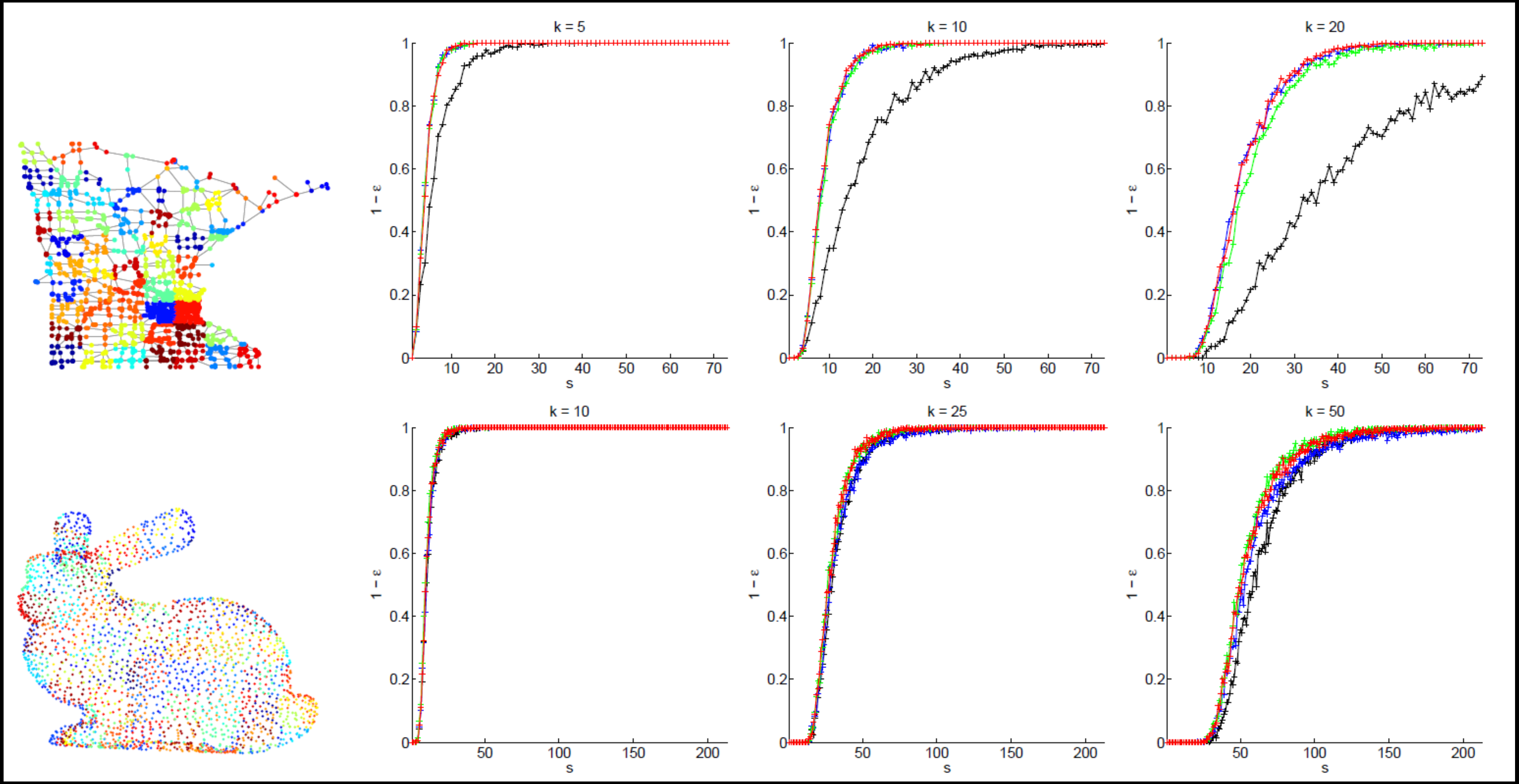
$$\|\mathsf{U}_k^\top \mathsf{U}_k \boldsymbol{z}^* - \boldsymbol{x}\| \leqslant \frac{1}{\sqrt{s(1-\delta)}} \left(A \left\| \tilde{\mathsf{P}}^{-1/2} \tilde{\boldsymbol{n}} \right\| + (B + \zeta E) \|\boldsymbol{x}\| \right)$$

$$\|\mathsf{U}_k^\top \mathsf{U}_k \boldsymbol{z}^* - \boldsymbol{z}^*\| \leqslant C \left\| \mathsf{P}^{-1/2} \boldsymbol{n} \right\| + D(1+\zeta)\|\boldsymbol{x}\|$$

$\forall \boldsymbol{x} \in \mathrm{span}(\mathsf{U}_k)$ and $\|\mathsf{A}^\top \mathsf{A}\boldsymbol{x} - \boldsymbol{x}\| \leqslant \zeta \|\boldsymbol{x}\|$

technicolor
FEEL THE WONDER

Empirical RIP



Group sampling distributions

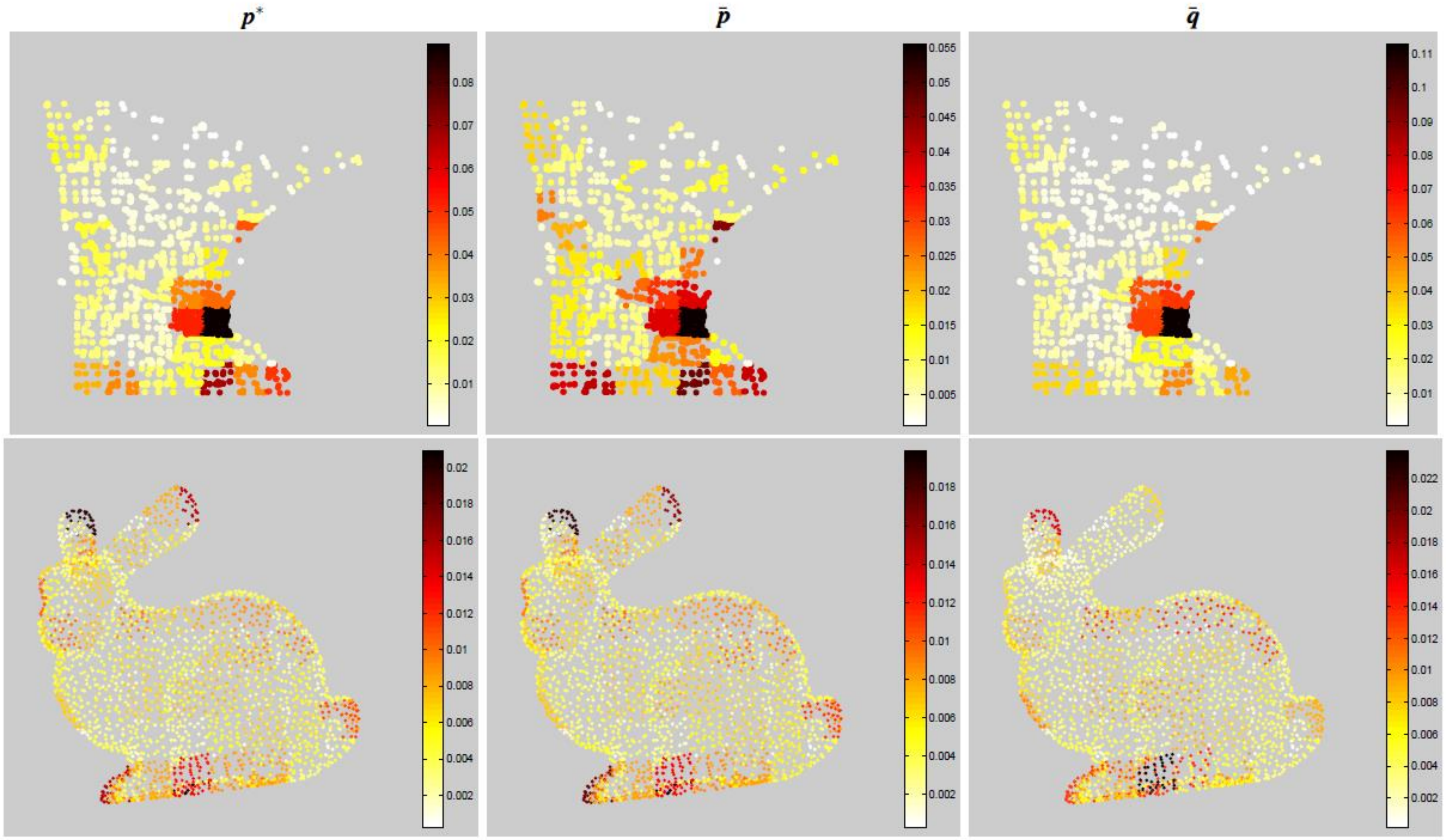
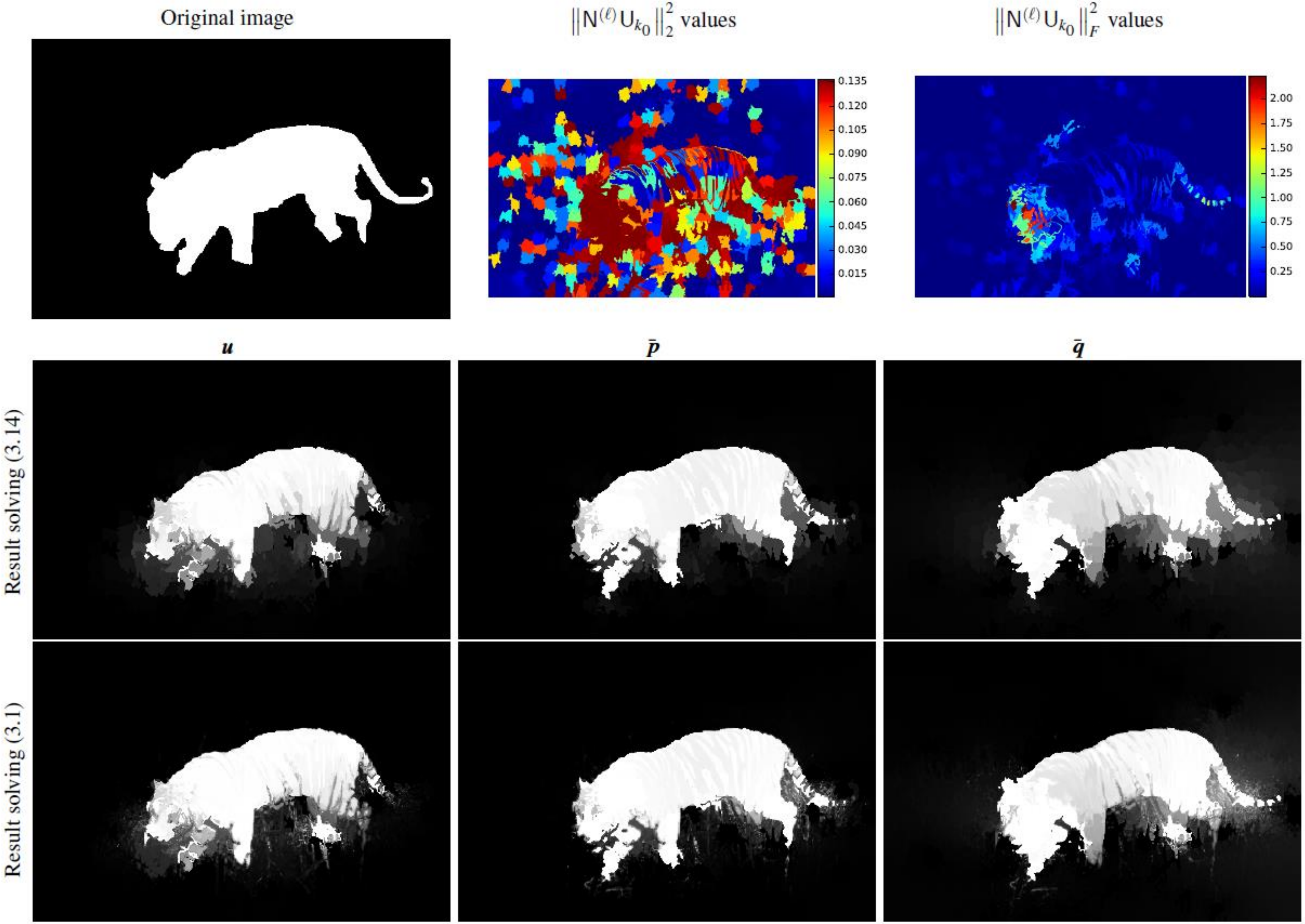


FIG. 4. Example of sampling distributions. Top panels: $\boldsymbol{p}^*$ (left), $\bar{\boldsymbol{p}}$ (middle), and $\bar{\boldsymbol{q}}$ (right) for the Minnesota graph at $k = 10$. Bottom panels: $\boldsymbol{p}^*$ (left), $\bar{\boldsymbol{p}}$ (middle), and $\bar{\boldsymbol{q}}$ (right) for the bunny graph at $k = 25$.

technicolor

FEEL THE WONDER



technicolor
FEEL THE WONDER

Convolutional Neural Nets (CNNs) on graph

CNNs

- Immensely successful for image-related task (recognition, prediction, processing, editing)
- Layers: Convolutions, non-linearities and pooling

Extension to graph signals?

- No natural convolution and pooling
- Graph structure may vary (not only size as with lattices)
- Computational complexity
- A simple proposal [Puy *et al*. 2017]

Graph-CNNs

Convolution in spectral domain [Bruna *et al.* 2013]

- Computation and use of Fourier basis not scalable
- Difficult handling of graph changes across inputs

Convolution with polynomial filters [Defferrard *et al*. 2016, Kipf *et al*. 2016]

- Better control of complexity and locality
- Not clear handling of graph changes across inputs
- Lack of filter diversity (e.g., rotation invariance on 2D lattice)

Direct convolutions [Monti *et al*. 2016, Niepert *et al*. 2016, Puy *et al*. 2017]

- Local or global pseudo-coordinates
- Include convolution on regular grid as special case

Direct convolution on weighted graph

[Puy *et al.* 2017]

At each vertex

- Extract a fixed-size signal "patch"

  Order,

  $$\sigma : \mathcal{V} \times (1, d) \to \mathcal{V}$$

  $\sigma(i, .)$ orders $\{j \in \mathcal{V} : j \sim i\}$

  Weigh,

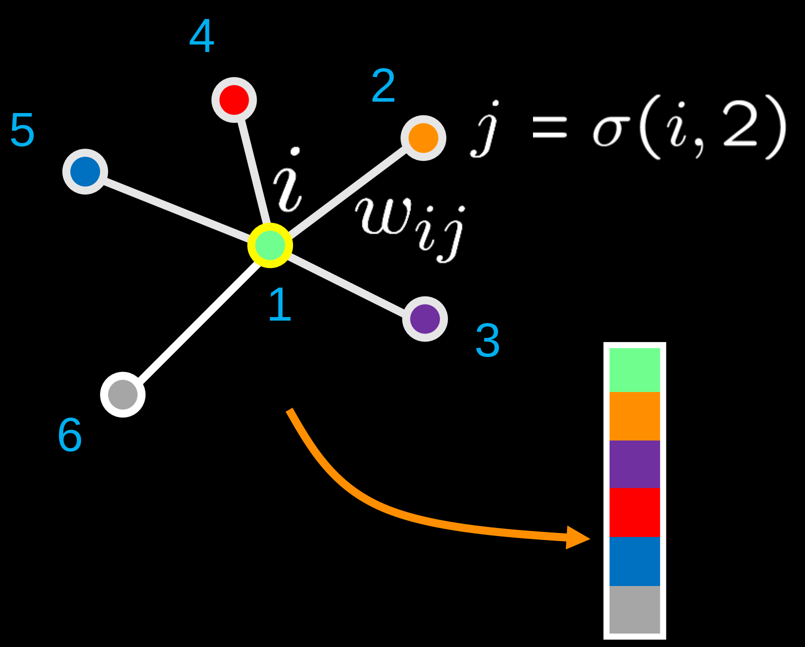
  $$g : \mathbb{R}^d_+ \times (1, d) \to \mathbb{R}_+$$

  Assemble

  $$q : \mathcal{V} \times \mathbb{R}^n \to \mathbb{R}^d$$
  $$(i, \boldsymbol{x}) \mapsto \left(g(\boldsymbol{w}_i, \ell) x_{\sigma(i,\ell)} \right)_{\ell=1}^d$$

- Dot product with filter kernel

  $$\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{h} \in \mathbb{R}^d$$
  $$\boldsymbol{x} \star \boldsymbol{h}(i) = \boldsymbol{h}^\top q(i, \boldsymbol{x})$$

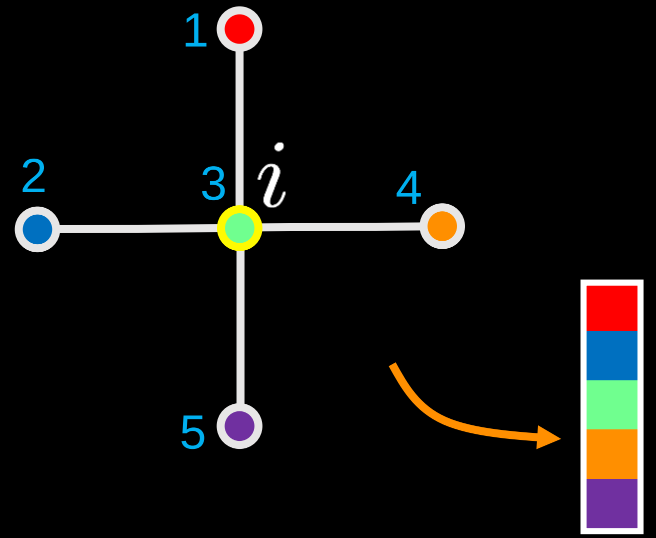


technicolor
FEEL THE WONDER

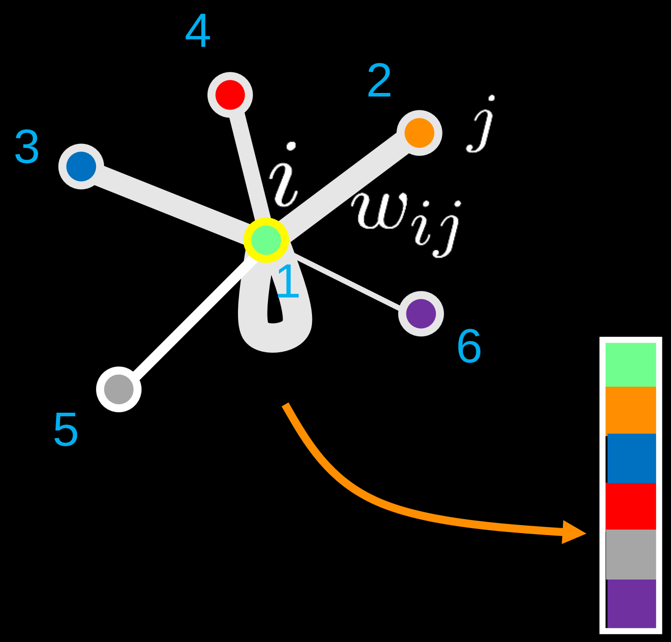
Direct convolution on weighted graph

Back to classic convolution

▶ Lexicographical order, no weighting



Weight-based ordering and weighting



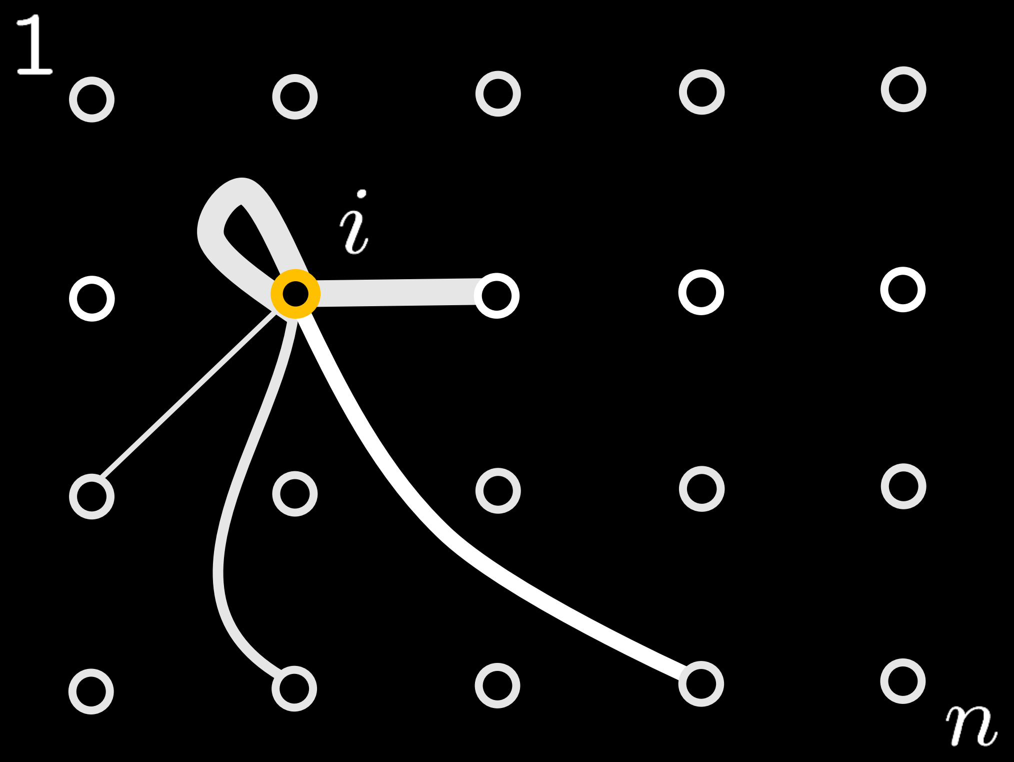
$$g(w_i, \ell) = \frac{w_{i\sigma(i,\ell)}}{\sum_{j \sim i} w_{ij}}$$

technicolor
FEEL THE WONDER

Non-local weighted pixel graph

Feature-based nearest neighbor graph

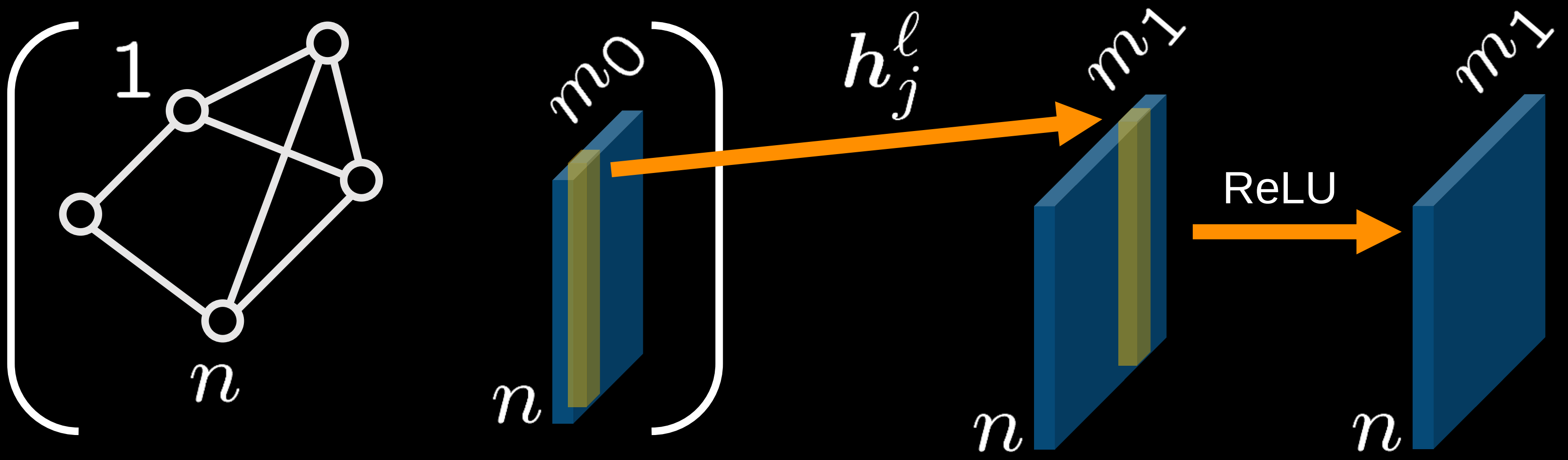
- Given an image, one feature vector at each pixel
- Connect each pixel to its *d* nearest neighbor in feature space
- Weigh with exponential of feature similarity



technicolor
FEEL THE WONDER

One graph convolutional layer

$$f(\mathcal{G}, \mathrm{X}_{n \times m_0}) = \left(\mathrm{ReLU}\Big(\sum_{j=1}^{m_0} \boldsymbol{x}_j \star \boldsymbol{h}_j^\ell + b_\ell\Big)\right)_{\ell=1\cdots m_1}$$



technicolor
FEEL THE WONDER

Style transfer

Neural example-based stylization [Gatys *et al*. 2015]

- Iterative modification of noise to fit “statistics” of style image and “content” of target image
- Neural statistics: Gram matrix of feature maps at a layer of a pre-trained deep CNN

Style transfer

$$n = 256 \times 256, d = 25$$
$$m_0 = 3, m_1 = 50$$

Using only a single random graph convolution layer

- Input image only used to build the graph

Style transfer

$$n = 256 \times 256, d = 25$$
$$m_0 = 3, m_1 = 50$$

Using only a single random graph convolution layer

- Input image only used to build the graph

Non-local graph only

Style transfer

$$n = 256 \times 256, d = 25$$
$$m_0 = 3, m_1 = 50$$

Using only a single random graph convolution layer

- Input image only used to build the graph

Non-local graph + Local graph

technicolor

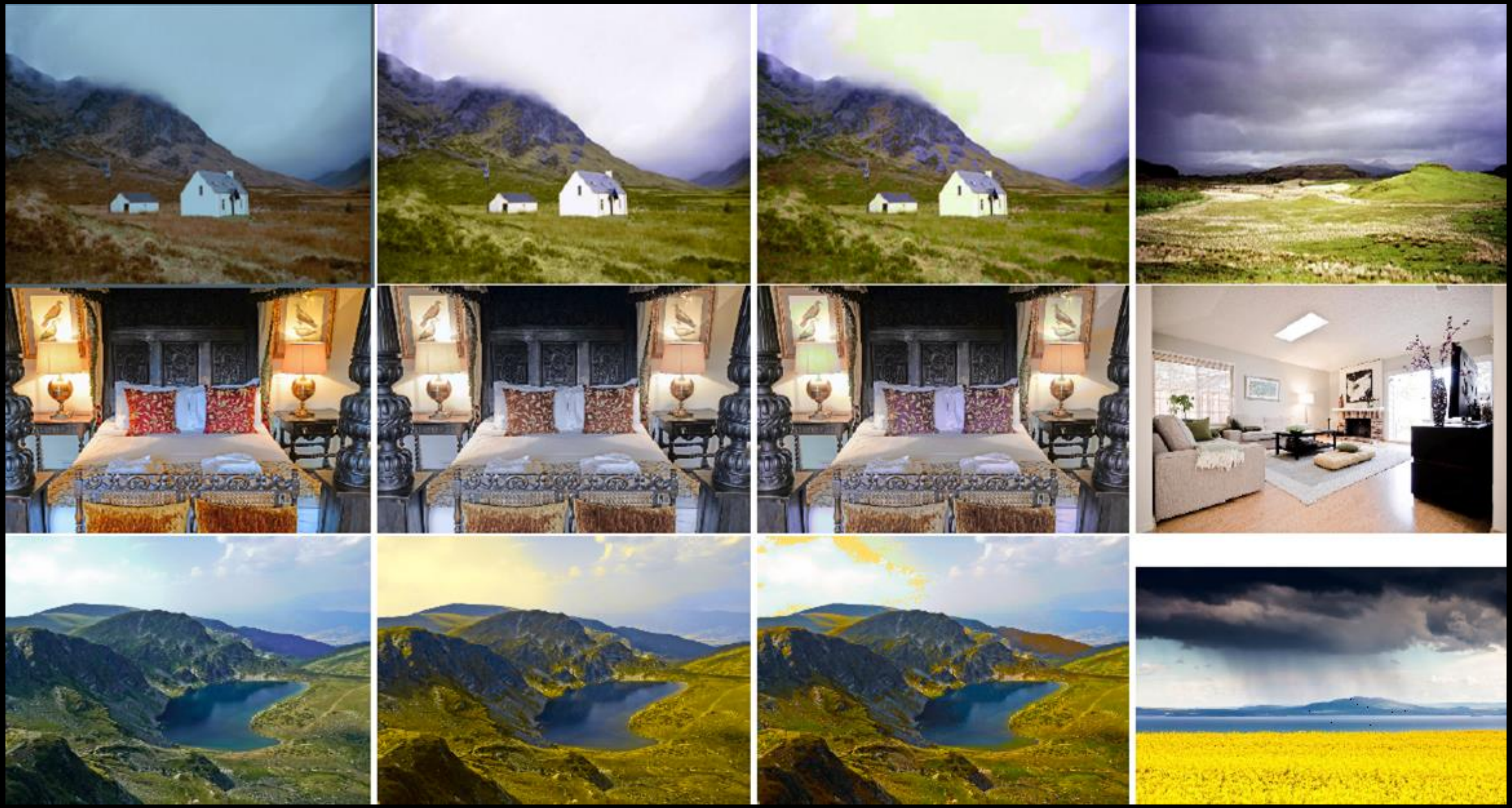
FEEL THE WONDER

$$n = 64, d = 10$$
$$m_0 = 2, m_1 = 100$$

Color palette transfer

Using only a single random graph convolution layer

target image | proposed | optimal transport | source palette



technicolor

FEEL THE WONDER

Signal denoising

Trained 3-layer graph CNN

- Local and non-local graphs from noisy input

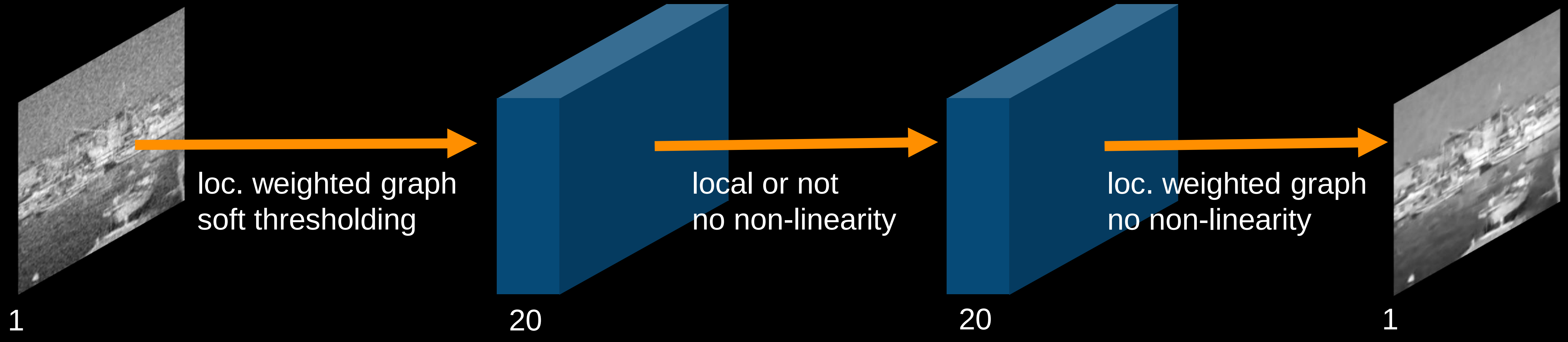
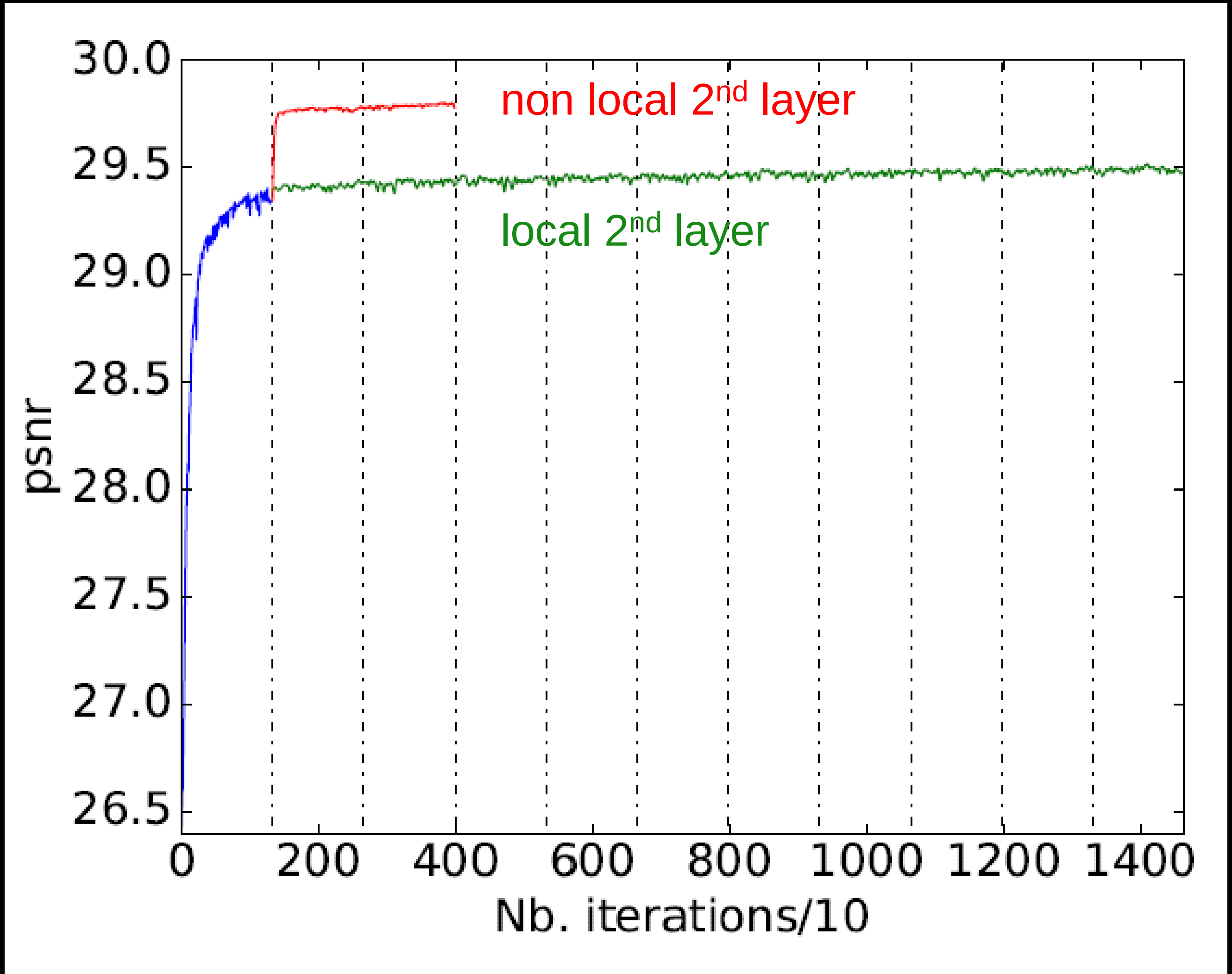


Image denoising



Noisy	23.10dB
Trained – Local	29.13dB
Trained – Non-local	29.42dB
Haar soft thresh.	26.78dB

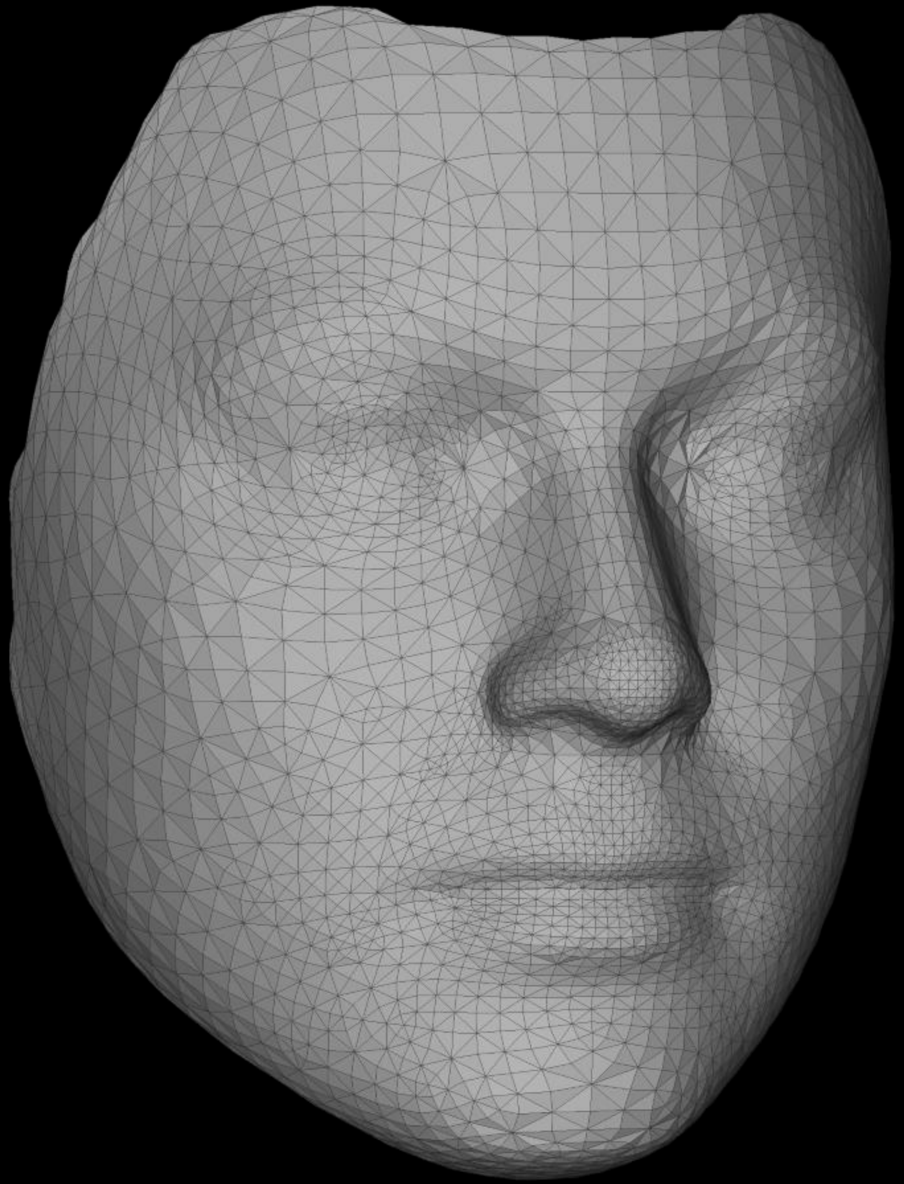
Triangular 3D mesh

Graph

- ► Vertices: points in 3D space
- ► Edges: forming triangulated graph
- ► Weights (if any): associated to local 3D shape

Signals

- ► Colors
- ► Normals
- ► Mesh deformations



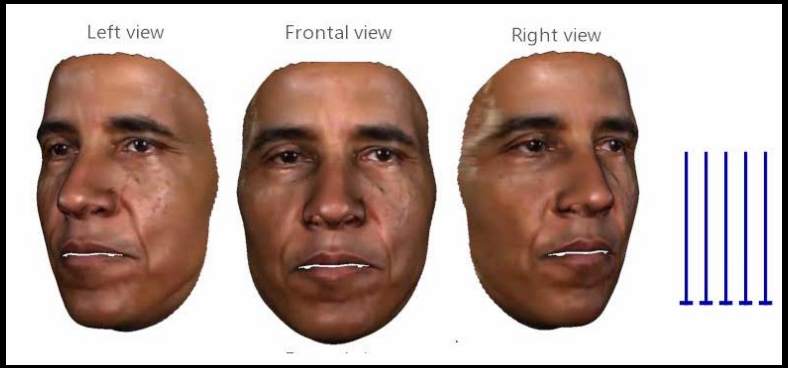
technicolor
FEEL THE WONDER

Face capture from single video

[Cao *et al.*, 2015]

[Suwajanakorn *et al.*, 2014]

[Garrido *et al.*, 2016]



Detailed 3D face rig

Parametric face model

Two-level coarse linear modelling

- Inter-individual variations: linear space around average neutral face (AAM)
- Expressions: linear space of main modes of deformations around neutral (*blendshapes*)

Reconstruction and tracking from raw measurements

- Extract person's neutral *shape* (morphology)
- Extract/track main *deformations* (expression/performance)
- Mitigate model limitations through *smooth corrections*
- Recover person-specific fine scale *details*

technicolor
FEEL THE WONDER

Parametric face model

Two-level coarse linear modelling

- ► Inter-individual variations: linear space around average neutral face (AAM)
- ► Expressions: linear space of main modes of deformations around neutral (*blendshapes*)

Reconstruction and tracking from raw measurements

- ► Extract person's neutral *shape* (morphology)
- ► Extract/track main *deformations* (expression/performance)
- ► Mitigate model limitations through *smooth corrections*
- ► Recover person-specific fine scale *details*

technicolor
FEEL THE WONDER

Parametric face model

Two-level coarse linear modelling

- Inter-individual variations: linear space around average neutral face (AAM)
- Expressions: linear space of main modes of deformations around neutral (*blendshapes*)

Reconstruction and tracking from raw measurements

- Extract person's neutral *shape* (morphology)
- Extract/track main *deformations* (expression/performance)
- Mitigate model limitations through *smooth corrections*
- Recover person-specific fine scale *details*

technicolor
FEEL THE WONDER

Parametric face model

Two-level coarse linear modelling

- Inter-individual variations: linear space around average neutral face (AAM)
- Expressions: linear space of main modes of deformations around neutral (*blendshapes*)

Reconstruction and tracking from raw measurements

- Extract person's neutral *shape* (morphology)
- Extract/track main *deformations* (expression/performance)
- Mitigate model limitations through *smooth corrections*
- Recover person-specific fine scale *details*

Smooth correction

[Garrido *et al*. 2016]

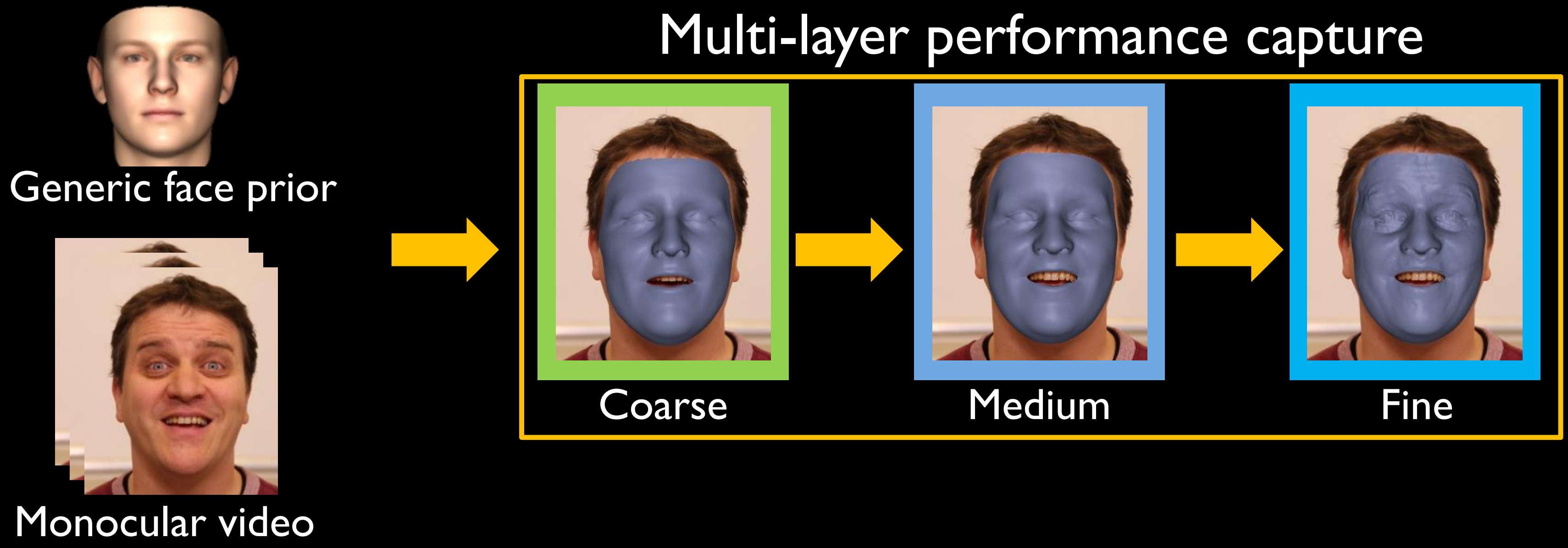
Layered mesh model

$$\boldsymbol{m} = \boldsymbol{m}_0 + \mathsf{A}\boldsymbol{\alpha} + \mathsf{B}\boldsymbol{\beta} + \mathsf{C}\boldsymbol{\eta} + \mathbf{d} \in \mathbb{R}^{3n}$$

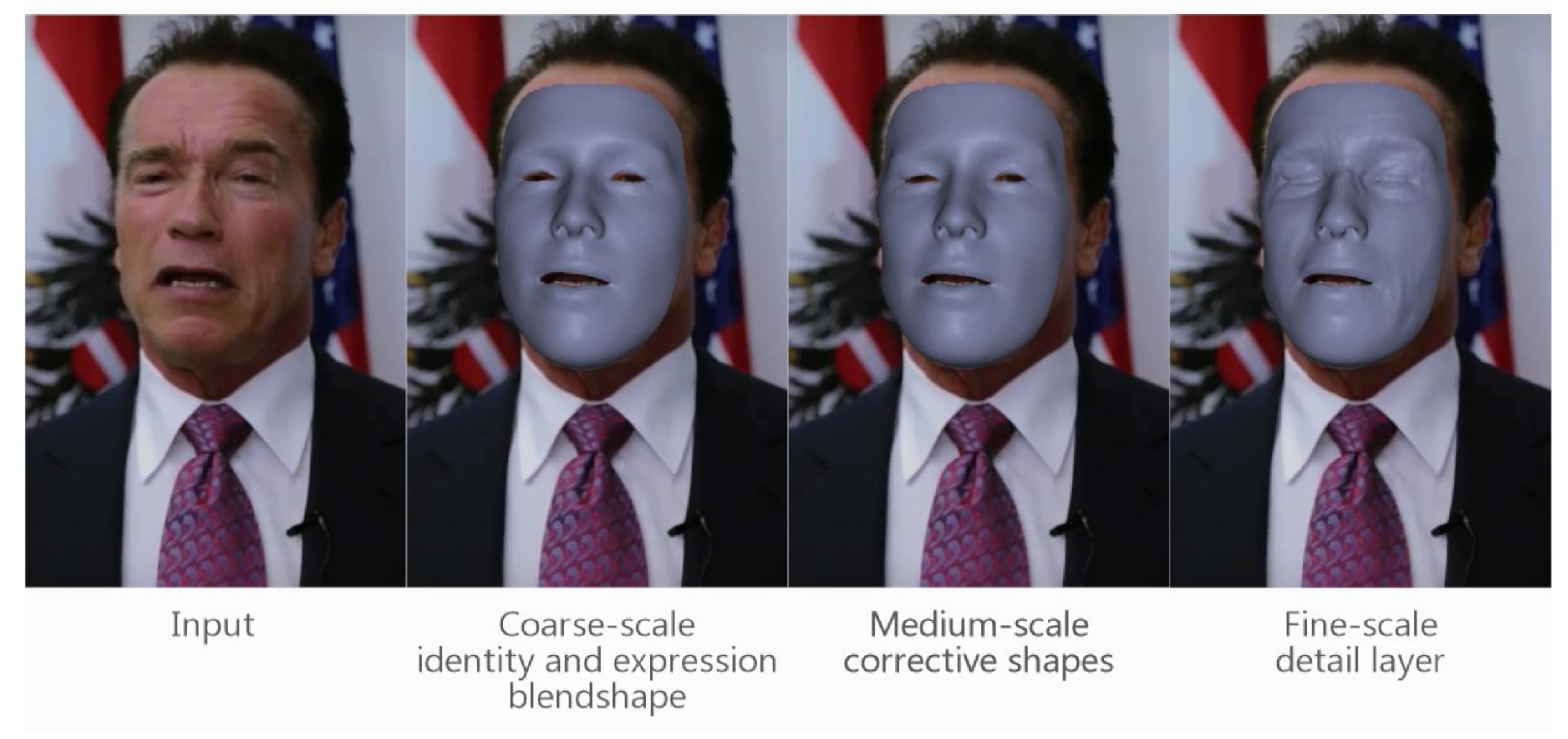
Graph harmonics on each coordinate [Vallet and Levy 2008][Li *et al*. 2013]

$$\mathsf{C}\boldsymbol{\eta} = \begin{bmatrix} \mathsf{U}_k\boldsymbol{\eta}_x \\ \mathsf{U}_k\boldsymbol{\eta}_y \\ \mathsf{U}_k\boldsymbol{\eta}_z \end{bmatrix}$$

Model personalization and tracking in single video

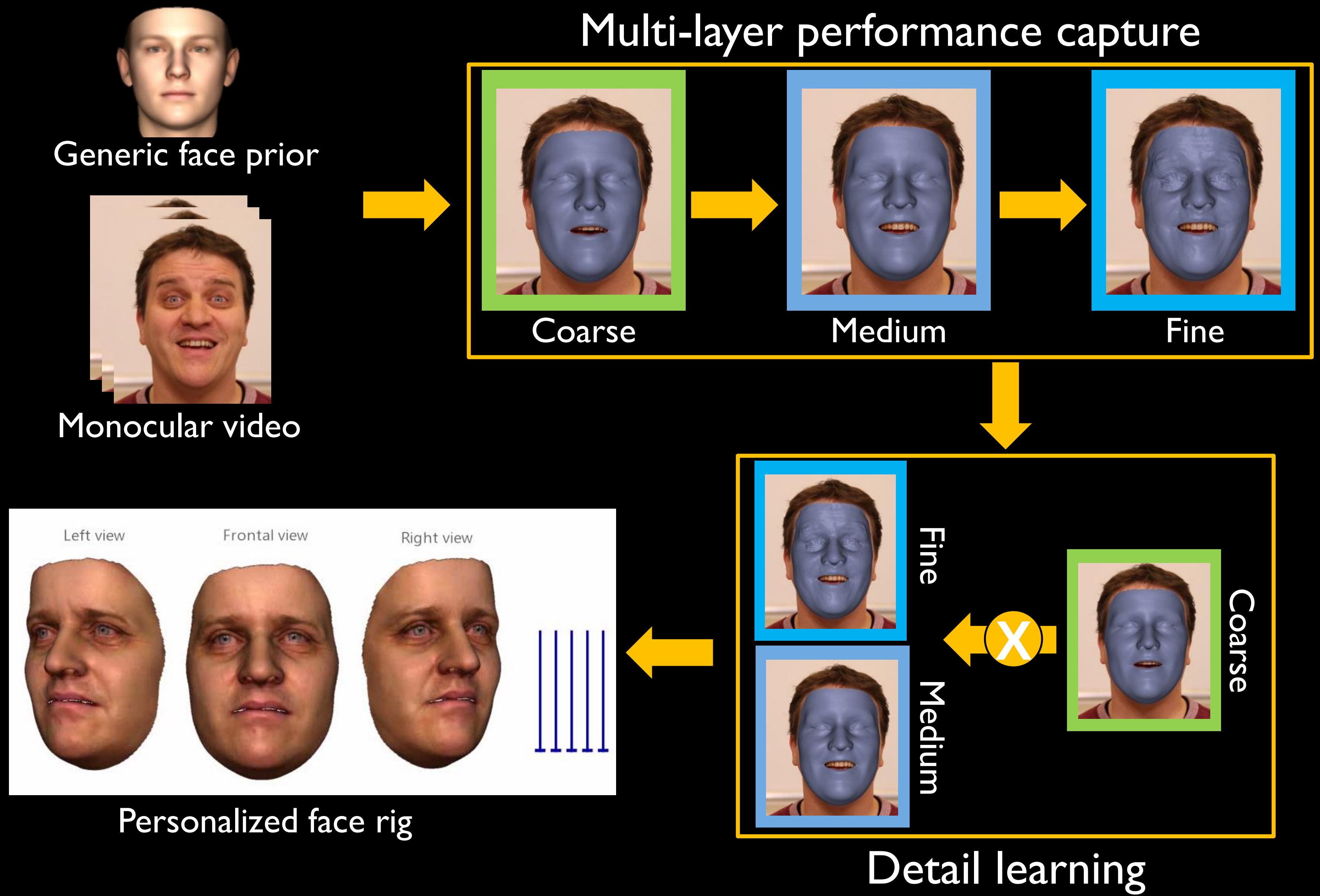


Multi-layer performance capture



technicolor
FEEL THE WONDER

From capture to animation

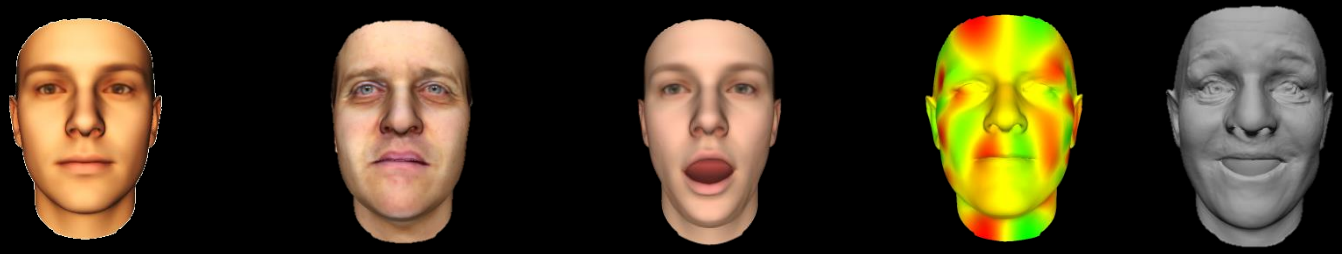


technicolor
FEEL THE WONDER

Personalized face rig

Turn model into a face rig (puppet)

$$m = m_0 + \mathrm{A}\alpha + \mathrm{B}\beta + \mathrm{C}\eta + \mathbf{d} \in \mathbb{R}^{3n}$$



- Ridge regression $\hat{\eta} = \mathrm{R}_{240\times 75}\beta$

technicolor

FEEL THE WONDER

Personalized face rig

Turn model into a face rig (puppet)

regression

fixed editable

$$\boldsymbol{m} = \boldsymbol{m}_0 + \mathsf{A}\boldsymbol{\alpha} + \mathsf{B}\boldsymbol{\beta} + \mathsf{C}\boldsymbol{\eta} + \mathbf{d} \in \mathbb{R}^{3n}$$

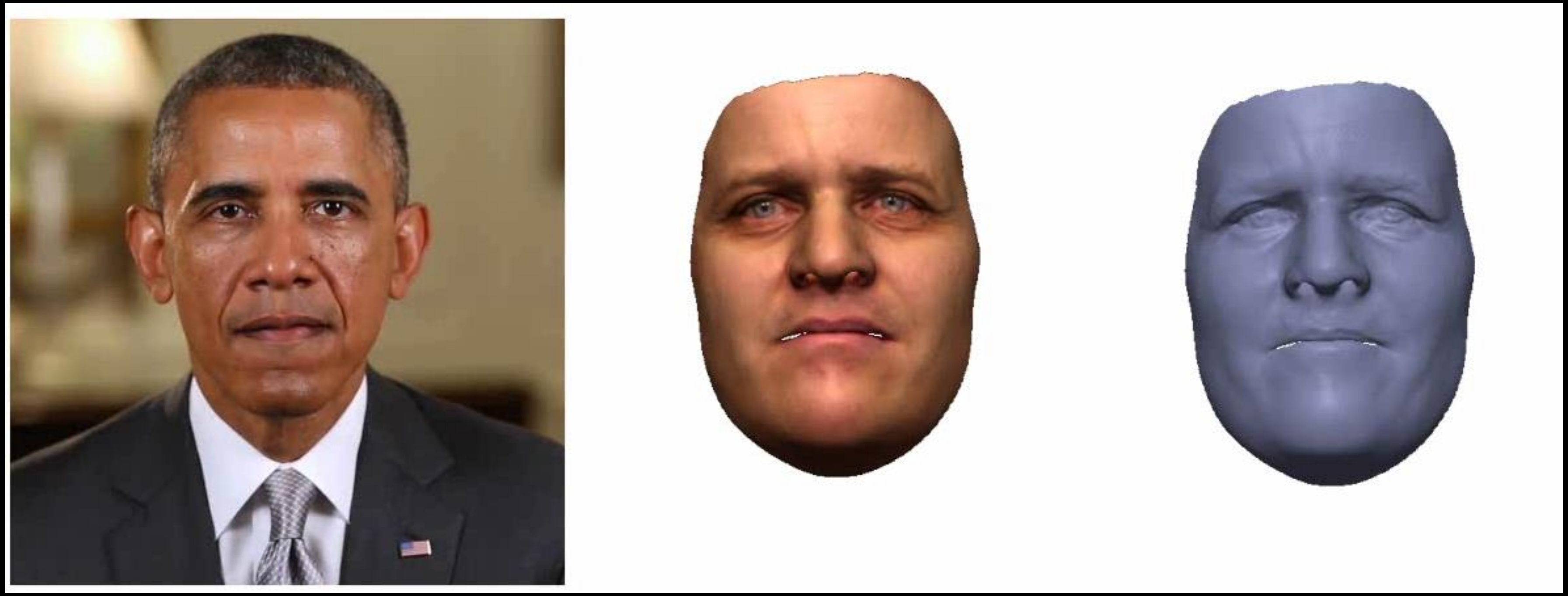
- Ridge regression $\hat{\boldsymbol{\eta}} = \mathsf{R}_{240\times 75}\boldsymbol{\beta}$

technicolor

FEEL THE WONDER

Rig animation from capture

Rig animation from capture



From processing to learning on graphs

Patrick Pérez

Maths and Images in Paris

IHP, 2 March 2017

- G. Puy, P. Pérez. Structured sampling and fast reconstruction of smooth graph signals. Submitted to Information and Inference
- G. Puy, S. Kitic, P. Pérez. Unifying local and non-local signal processing with graph CNNs. arXiv:1702.07759
- P. Garrido, M. Zollhoefer, D. Casas, L. Valgaerts, K. Varanasi, P. Pérez, Ch. Theobalt. Reconstruction of personalized 3D face rigs from monocular video. ACM Trans. on Graghics, 35(3), 2016

technicolor

FEEL THE WONDER