NYU COURANT INSTITUTE OF MATHEMATICAL SCIENCES
NYU CENTER FOR DATA SCIENCE

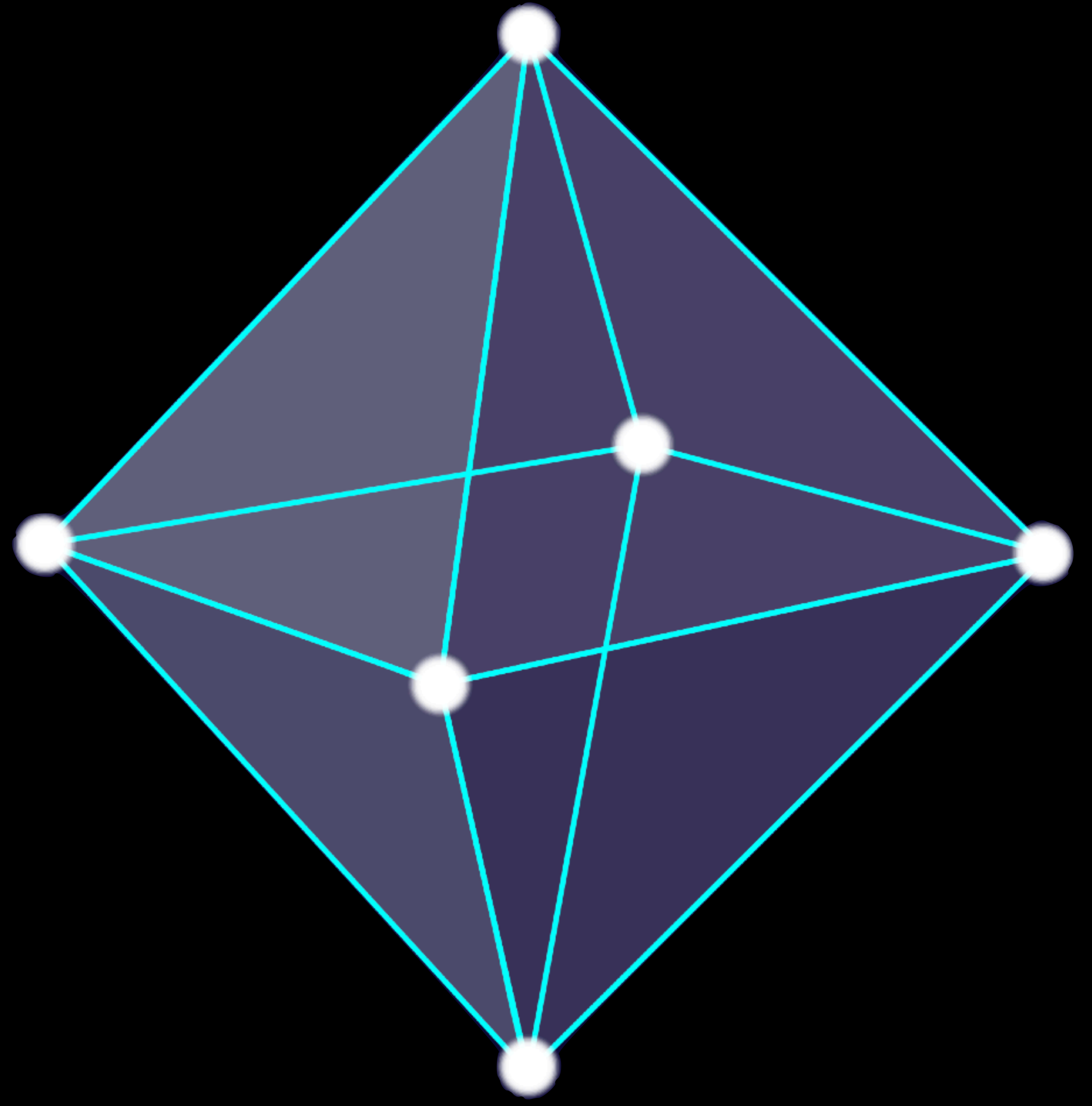
JOAN BRUNA

ON SPARSE LINEAR PROGRAMMING AND NEURAL NETWORKS

joint work with

Jaume de Dios
(UCLA)

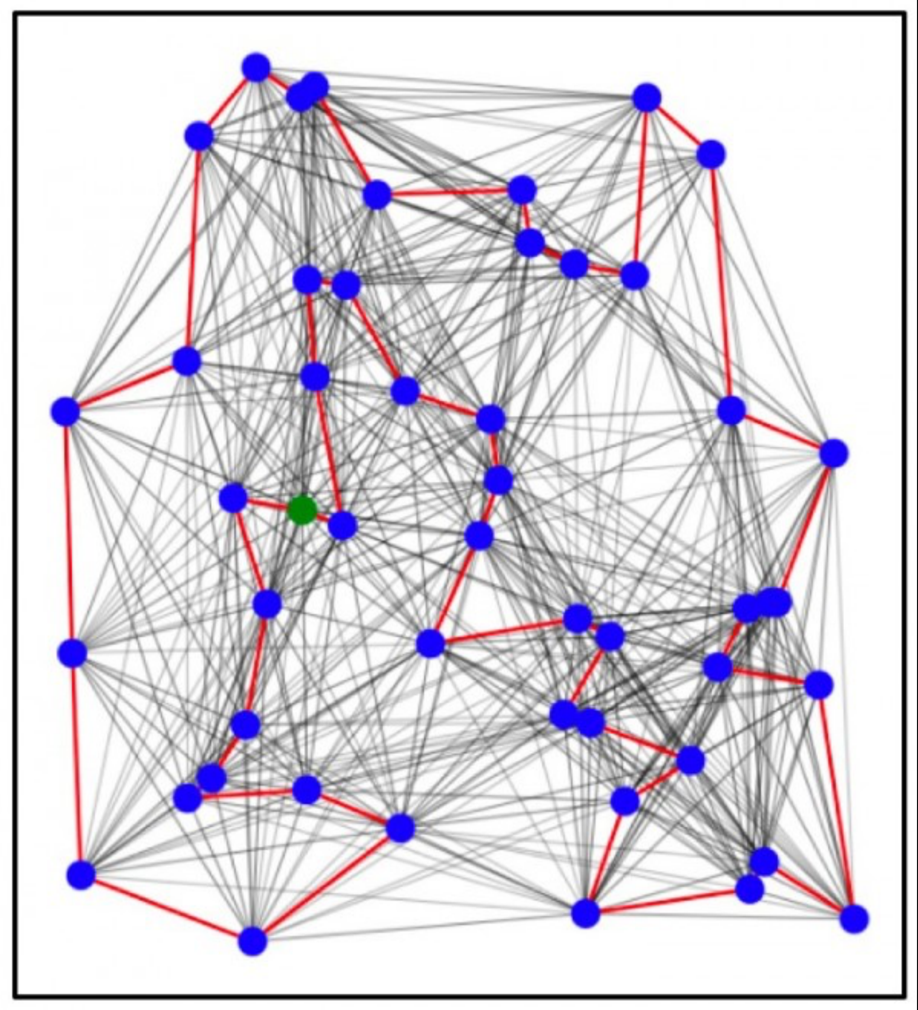
Luca Venturi
(NYU)

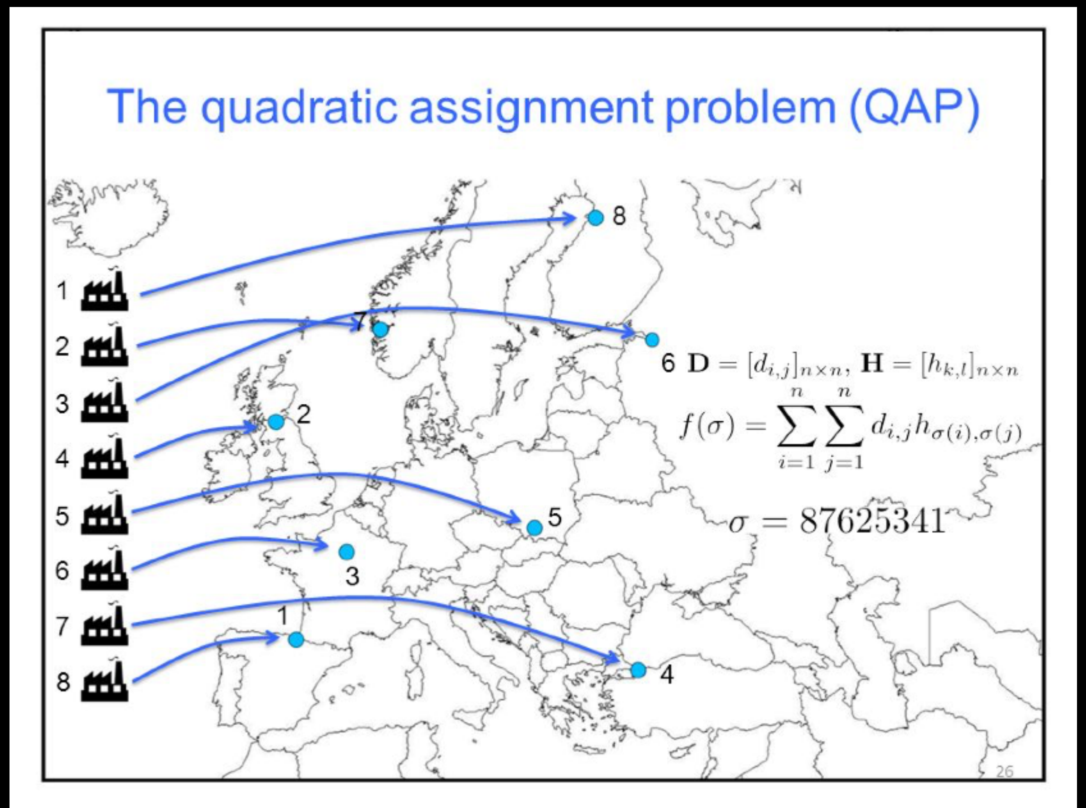


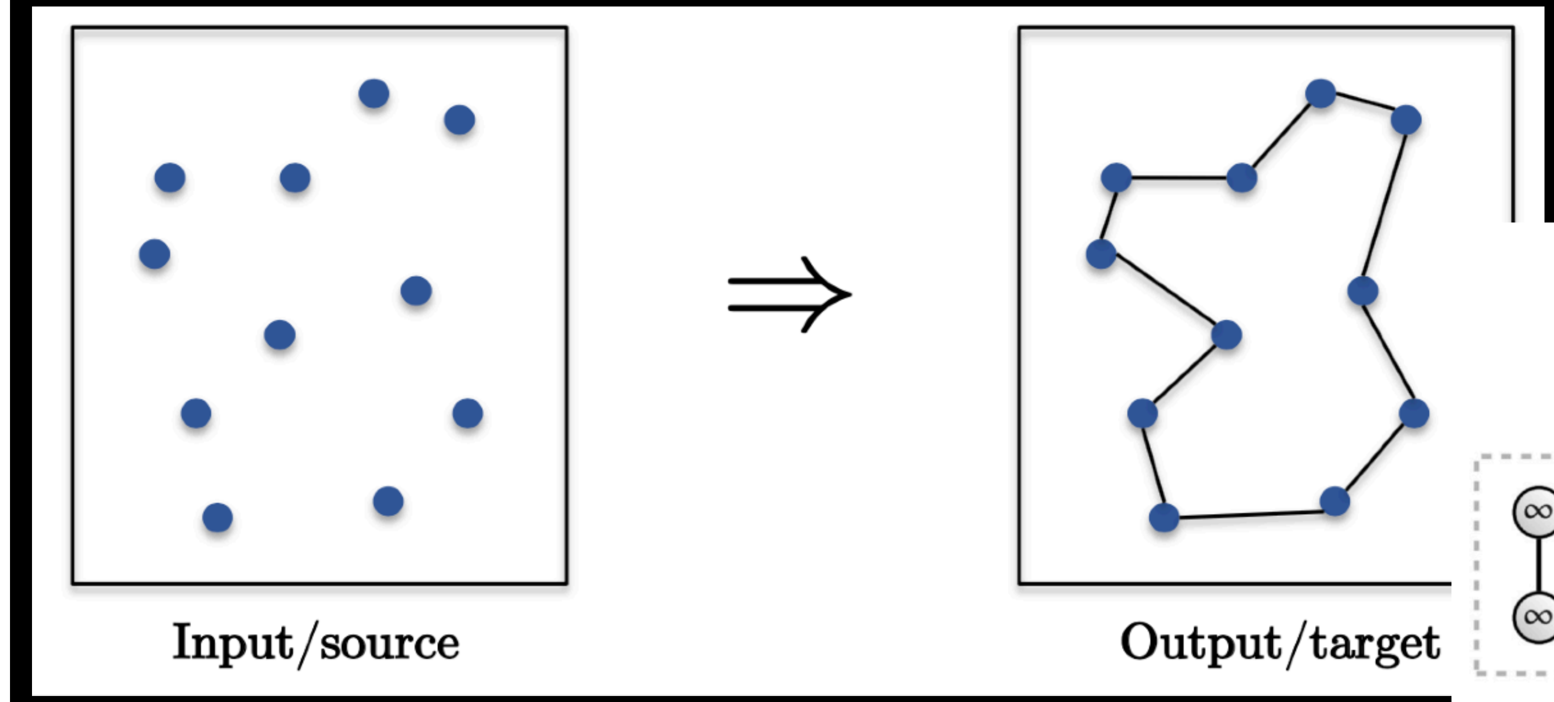
(Fig credit: John Cook)

DEEP LEARNING AND COMBINATORIAL OPTIMIZATION

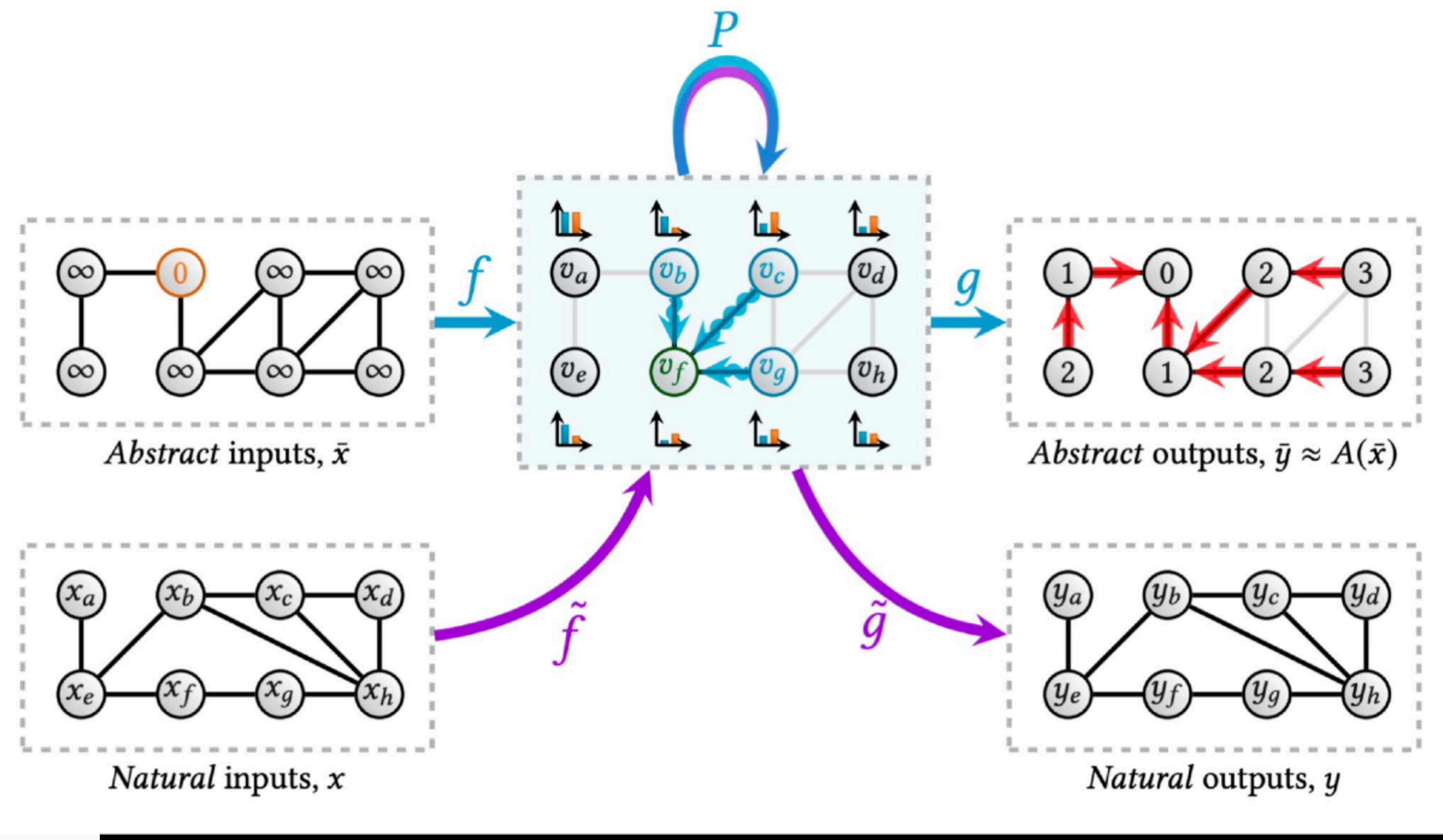
- What can DL do for CO?



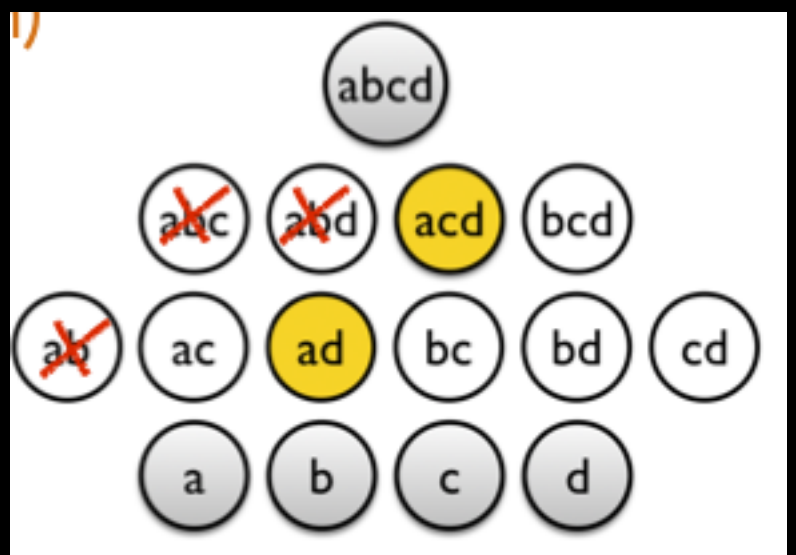




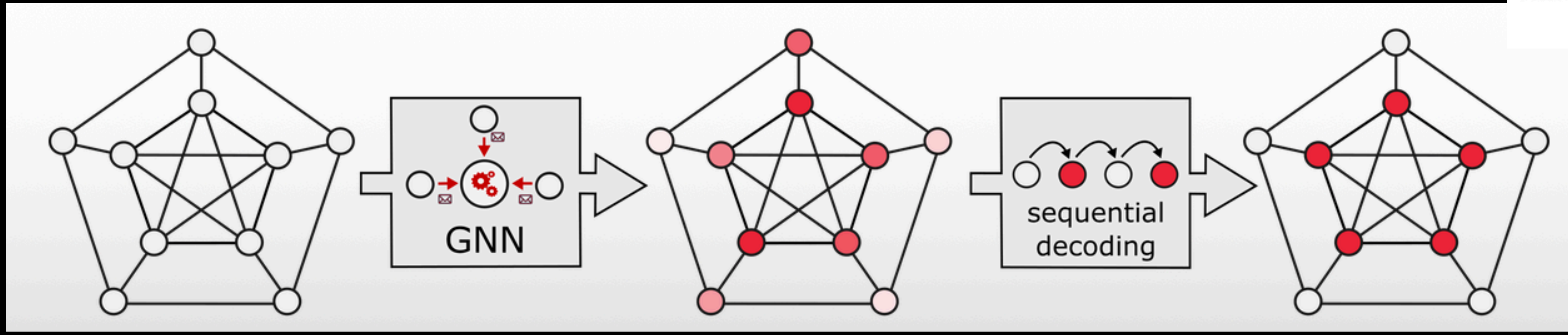
[X. Bresson's talk]



[Petar Veličković]



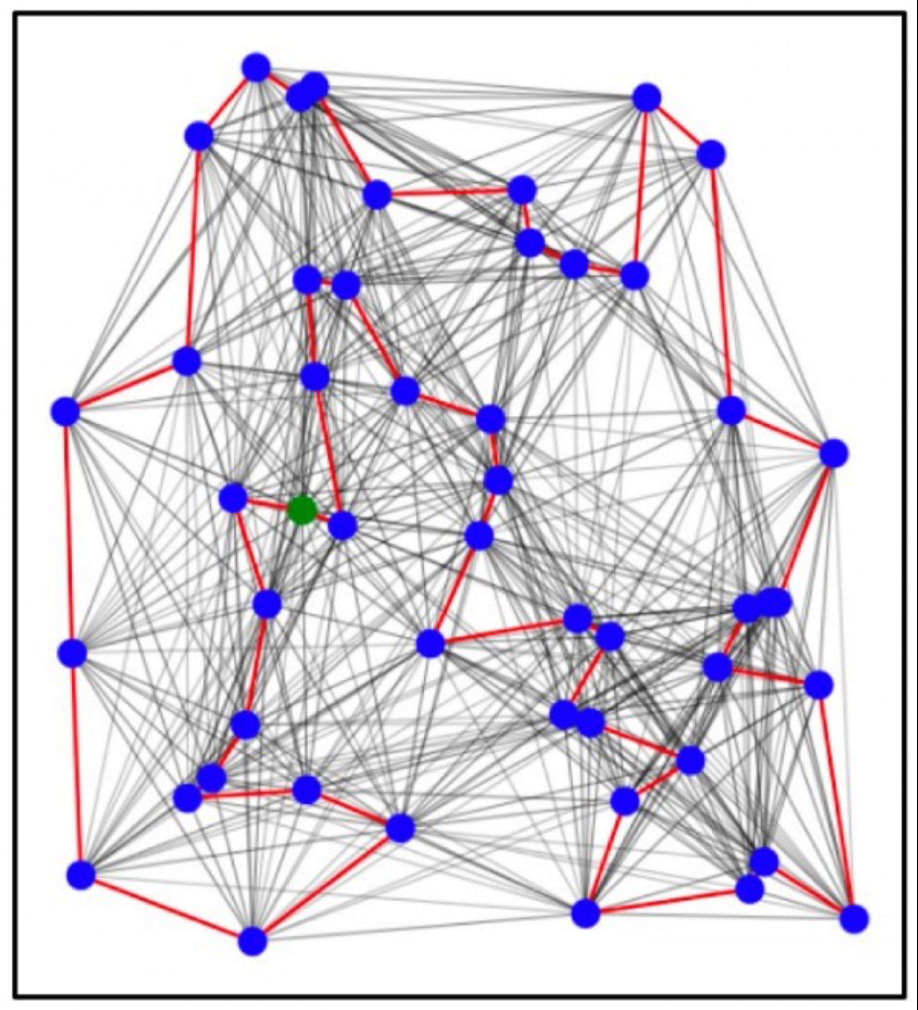
[Kyle Cranmer]

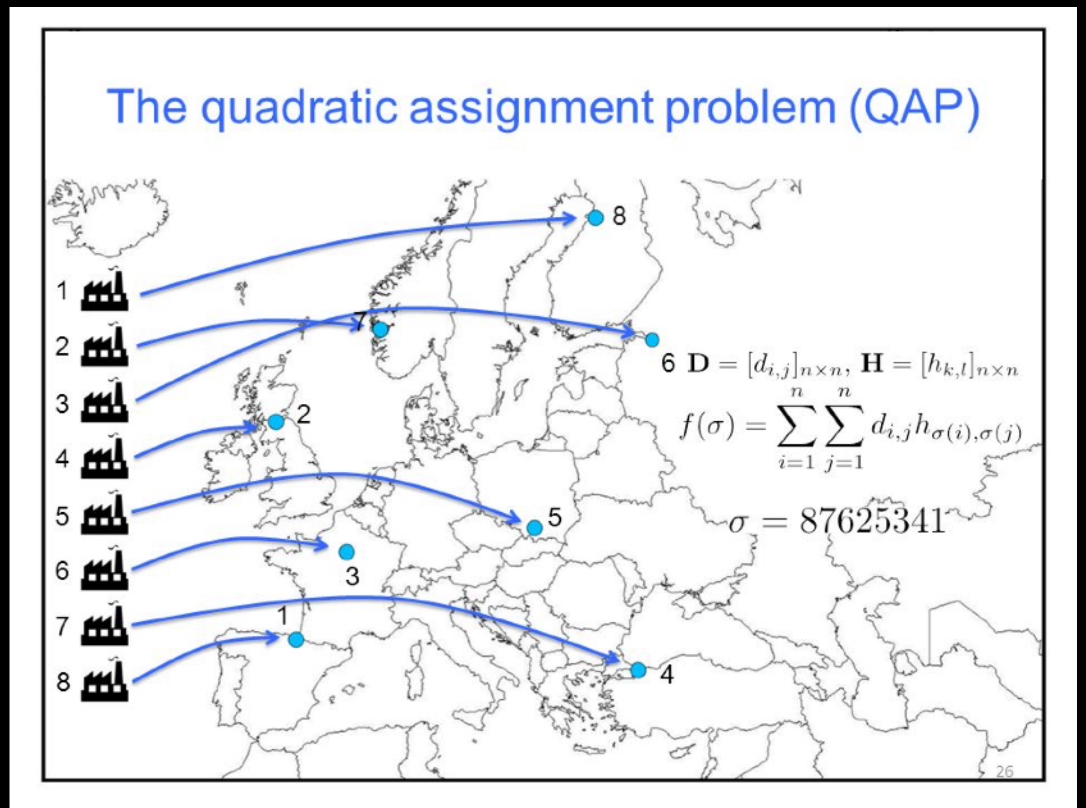


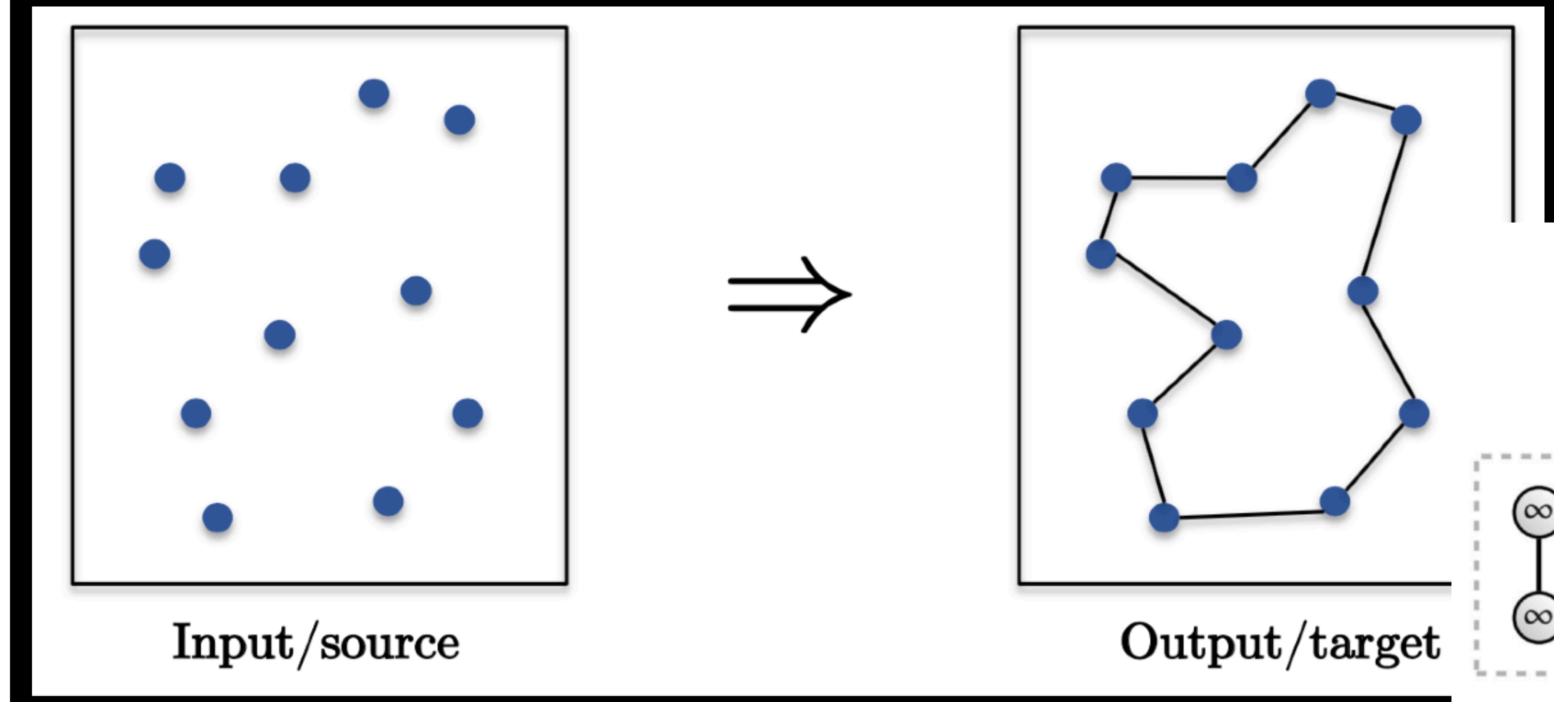
[Andreas Loukas]

DEEP LEARNING AND COMBINATORIAL OPTIMIZATION

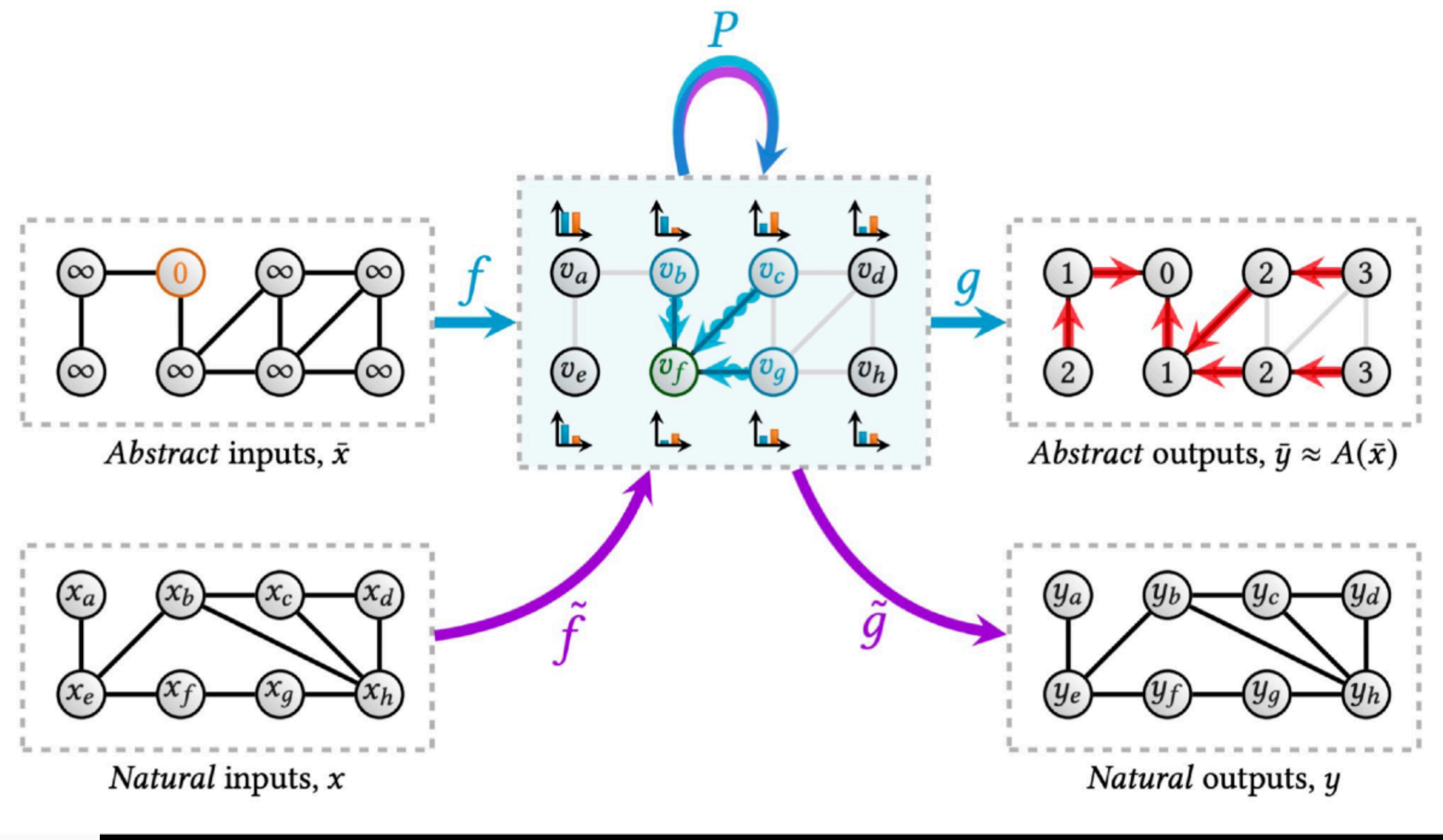
- What can DL do for CO?



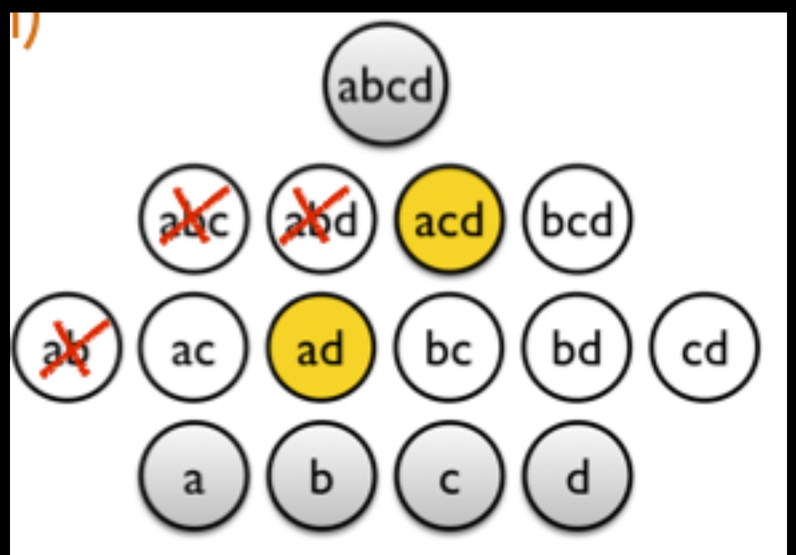




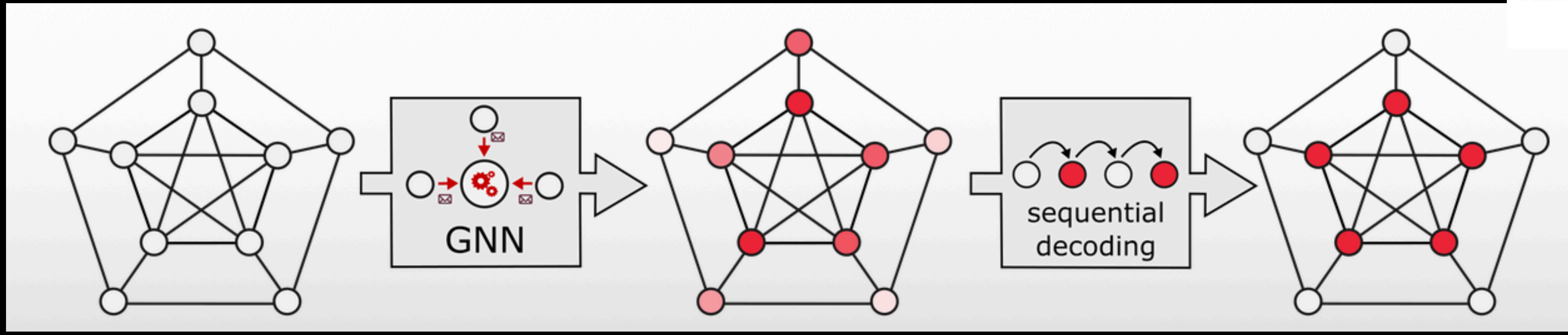
[X. Bresson's talk]



[Petar Veličković]



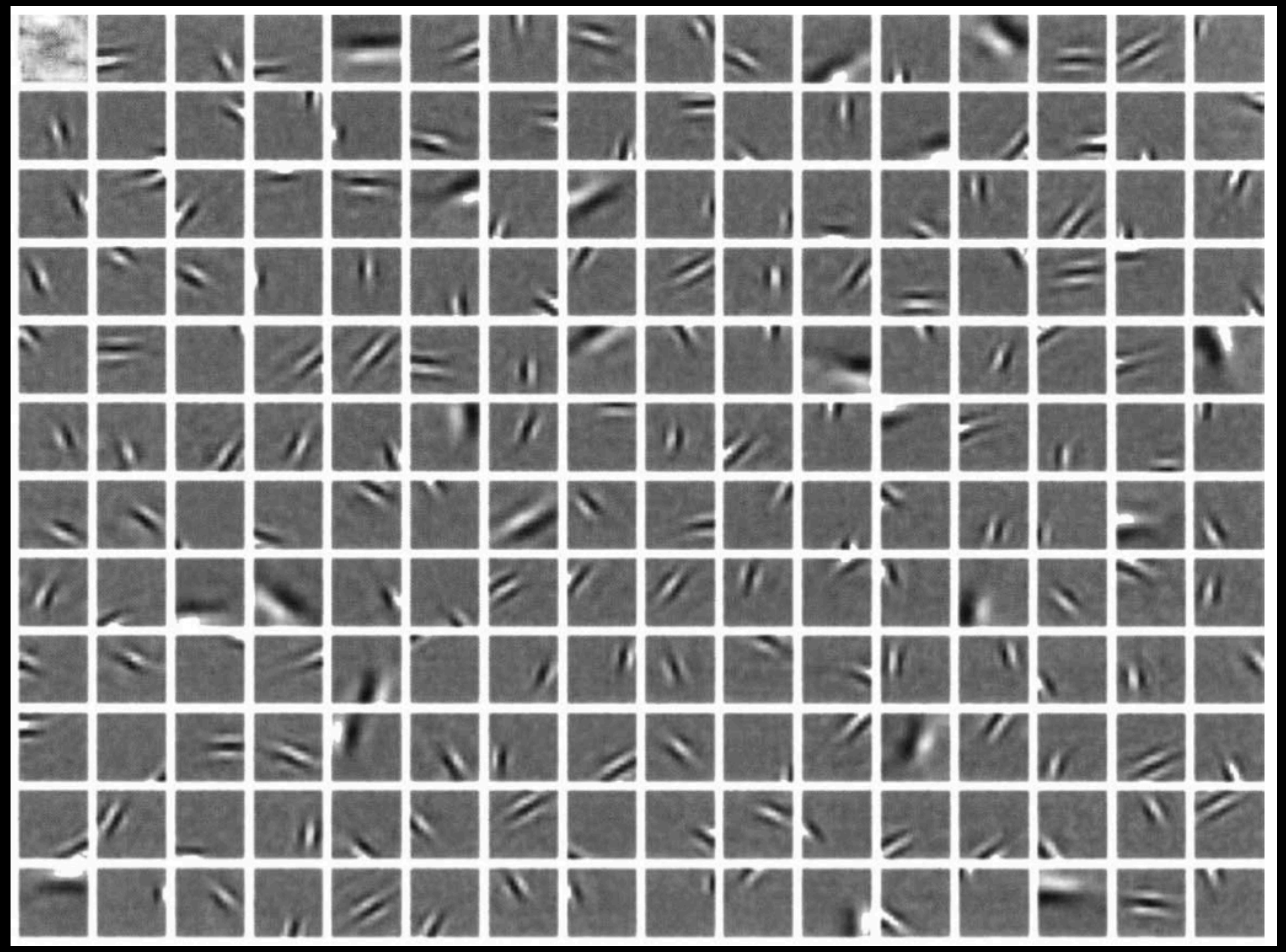
[Kyle Cranmer]



[Andreas Loukas]

- What can CO do for DL?

SPARSE INFERENCE



[Olshausen & Field]

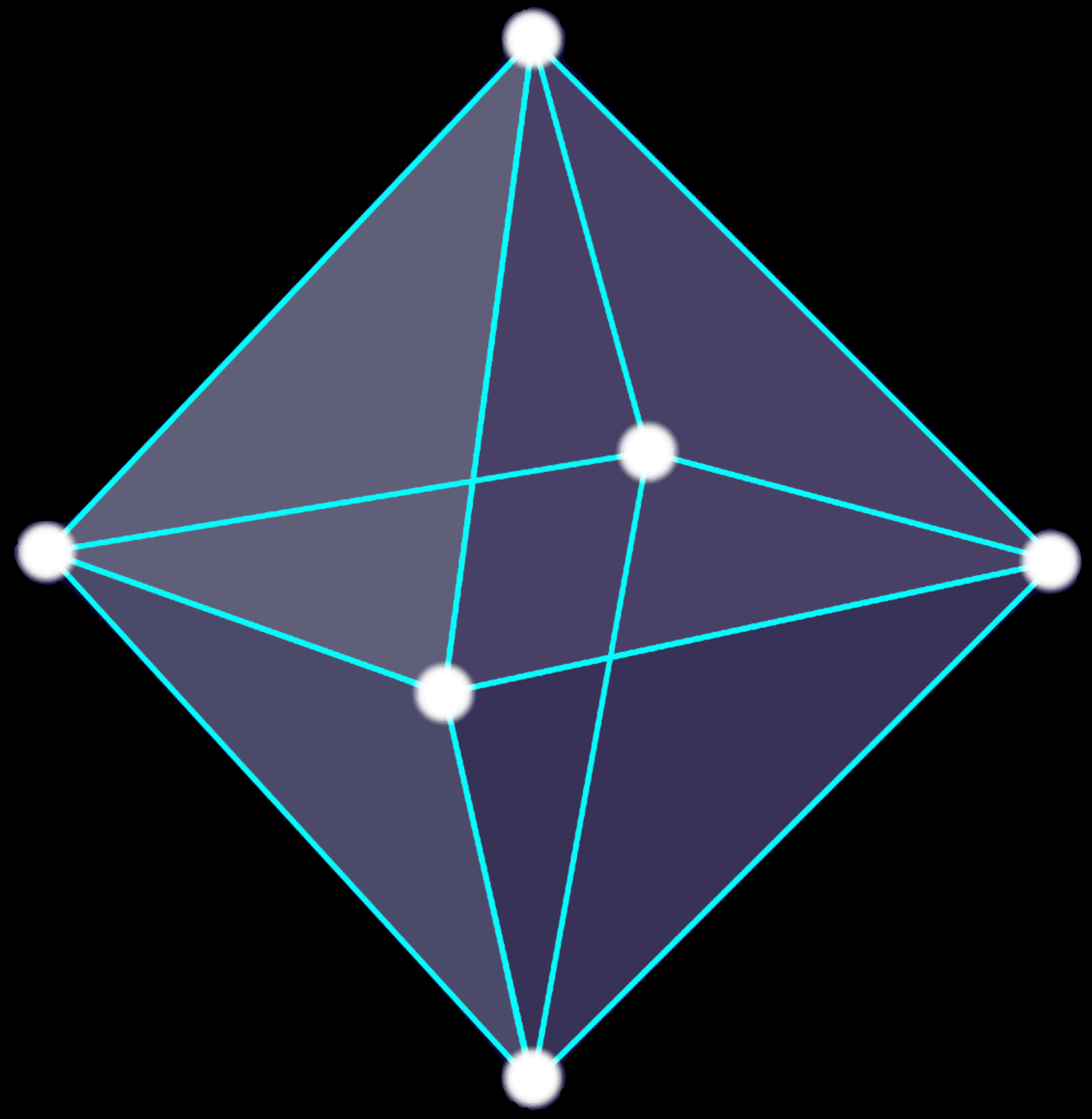
- Sparse Linear Recovery: Canonical Template for Combinatorial Optimization [Natarajan]:

  - Given dictionary $W \in \mathbb{R}^{d\times m}$, $m > d$, and $x = Wz$, recover z by exploiting a sparsity prior.

$$f_W^*(x) := \arg\min \{\|z\|_0;\ x = Wz\}\,.$$

  - Basic framework to understand/analyse power of nonlinear approximation relative to linear approximation [DeVore].

SPARSE INFERENCE

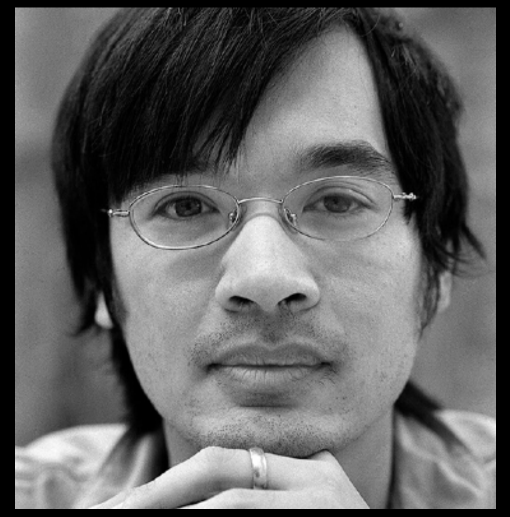


- Sparse Linear Recovery: Canonical Template for Combinatorial Optimization [Natarajan]:
  - Given dictionary $W \in \mathbb{R}^{d \times m}$, $m > d$, and $x = Wz$, recover z by exploiting a sparsity prior.

$$f_W^*(x) := \arg\min \{ \|z\|_0 ;\ x = Wz \} .$$

  - Basic framework to understand/analyse power of nonlinear approximation relative to linear approximation [DeVore].
- Convex Relaxation: replace ℓ_0 with ℓ_1 norm.
  - Compressed Sensing [Candes, Romberg, Tao, Donoho]
  - Efficient Algorithms leveraging convex geometry.






THIS TALK: SPARSE INFERENCE MEETS NEURAL NETWORKS

- ***Memorization in Overparametrised Shallow Networks***
  - Given dataset $\{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i \leq n}$, find “smallest” shallow net $f(\cdot, \Theta^*)$ such that $f(x_i, \Theta^*) = y_i$, $i \in [n]$.
  - Guarantees in the Mean-Field infinitely wide limit back to finite-width?

THIS TALK: SPARSE INFERENCE MEETS NEURAL NETWORKS

- ***Memorization in Overparametrised Shallow Networks***

  - Given dataset $\{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i \leq n}$, find "smallest" shallow net $f(\cdot, \Theta^*)$ such that $f(x_i, \Theta^*) = y_i \, , i \in [n]$.
  - Guarantees in the Mean-Field infinitely wide limit back to finite-width?

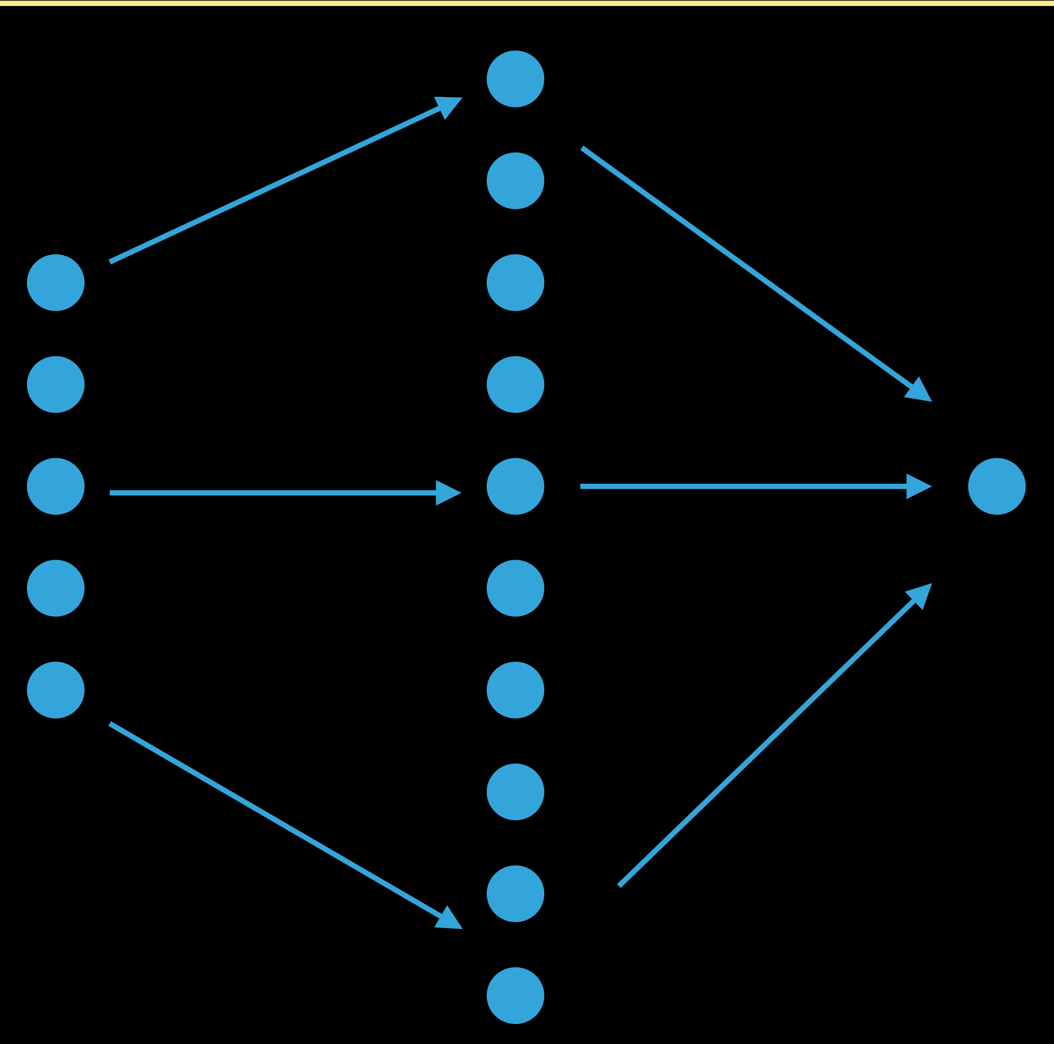
- ***Neural function approximation of sparse inference***

  - Given high-dimensional input $x \in \mathbb{R}^d$ and dictionary $W \in \mathbb{R}^{d \times m}$, sparse regression defined as $f^*_W(x) := \arg\min \{\|z\|_0; \; x = Wz\}$.
  - Neural network approximation of f^*_W?
  - In particular, is depth needed in the high-dimensional regime?

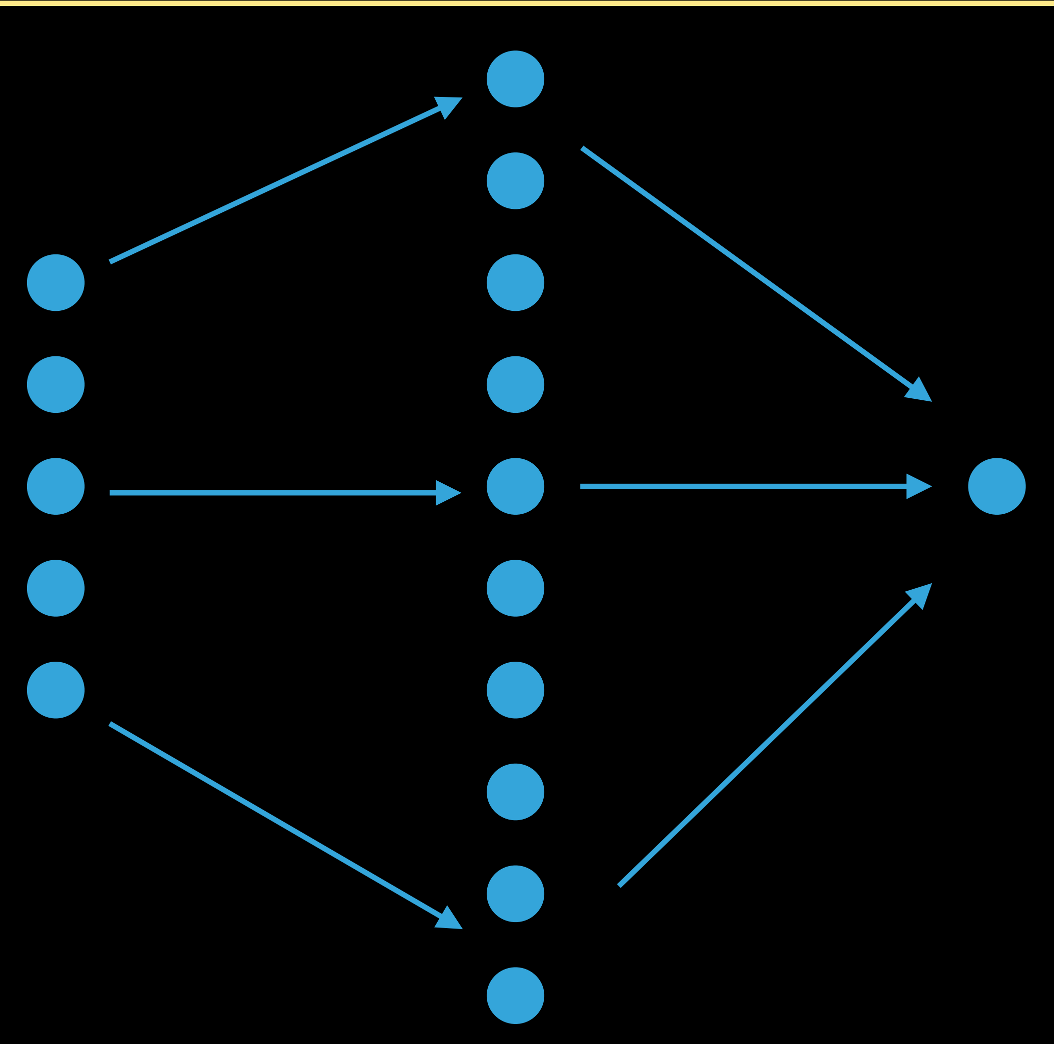
MEMORIZATION IN SHALLOW NEURAL NETWORKS: SET-UP

- Single hidden-layer ReLU network with input in $\mathbb{R}^d$ and parameters $\Theta = \{\theta_j = (a_j, b_j, c_j) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}\}_{j=1}^M$:

$$f(x;\Theta) = \frac{1}{M}\sum_{j=1}^{M} c_j(a_j^\top x + b_j)_+ \ .$$



MEMORIZATION IN SHALLOW NEURAL NETWORKS: SET-UP

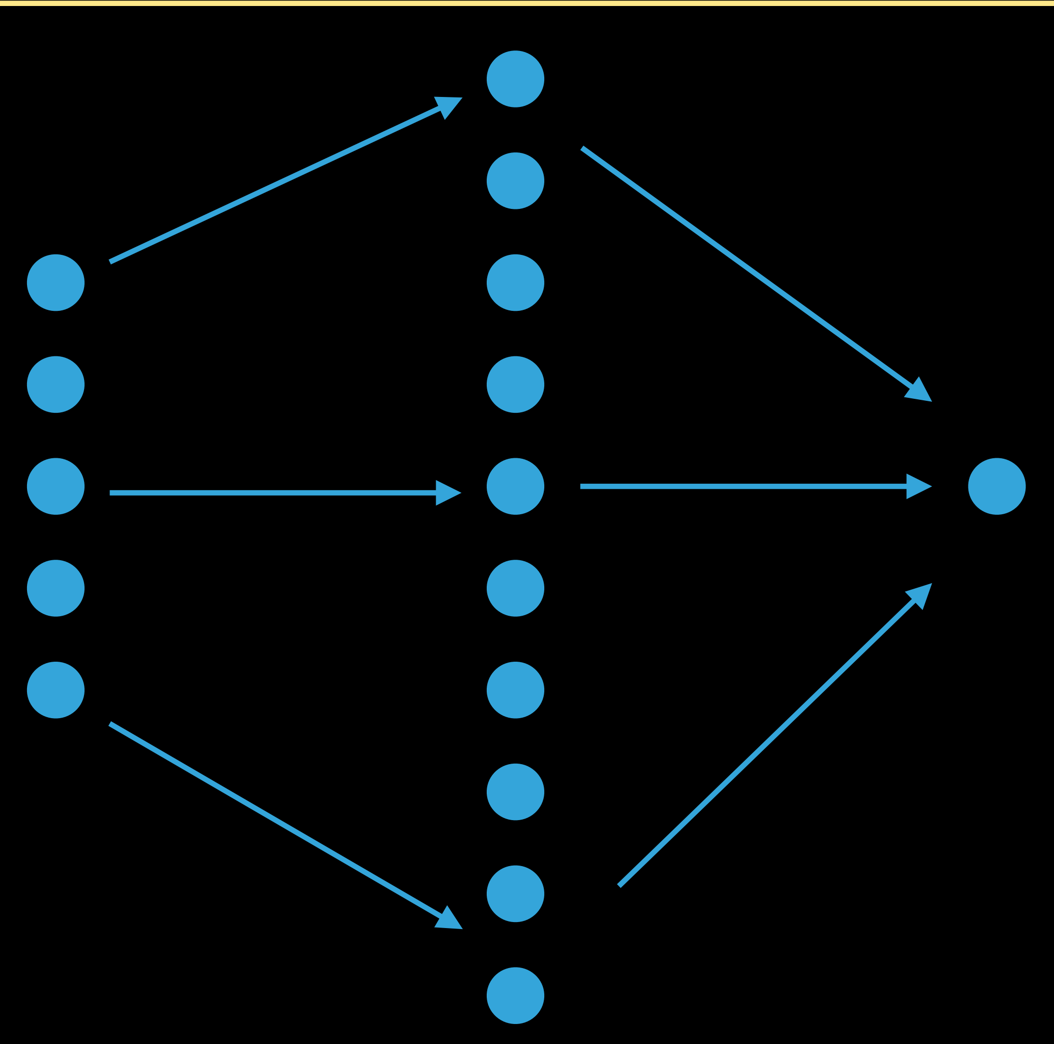


- Single hidden-layer ReLU network with input in $\mathbb{R}^d$ and parameters $\Theta = \{\theta_j = (a_j, b_j, c_j) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}\}_{j=1}^M$:

$$f(x;\Theta) = \frac{1}{M}\sum_{j=1}^{M} c_j (a_j^\top x + b_j)_+ \ .$$

- Goal: Memorize training set $\{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i \le n}$, ie find Θ^* such that $f(x_i;\Theta^*) = y_i$, with ***small*** complexity, e.g. smallest possible M, or smallest weights $\frac{1}{M}\sum_{j=1}^{M}\|\theta_j\|^2$

MEMORIZATION IN SHALLOW NEURAL NETWORKS: SET-UP

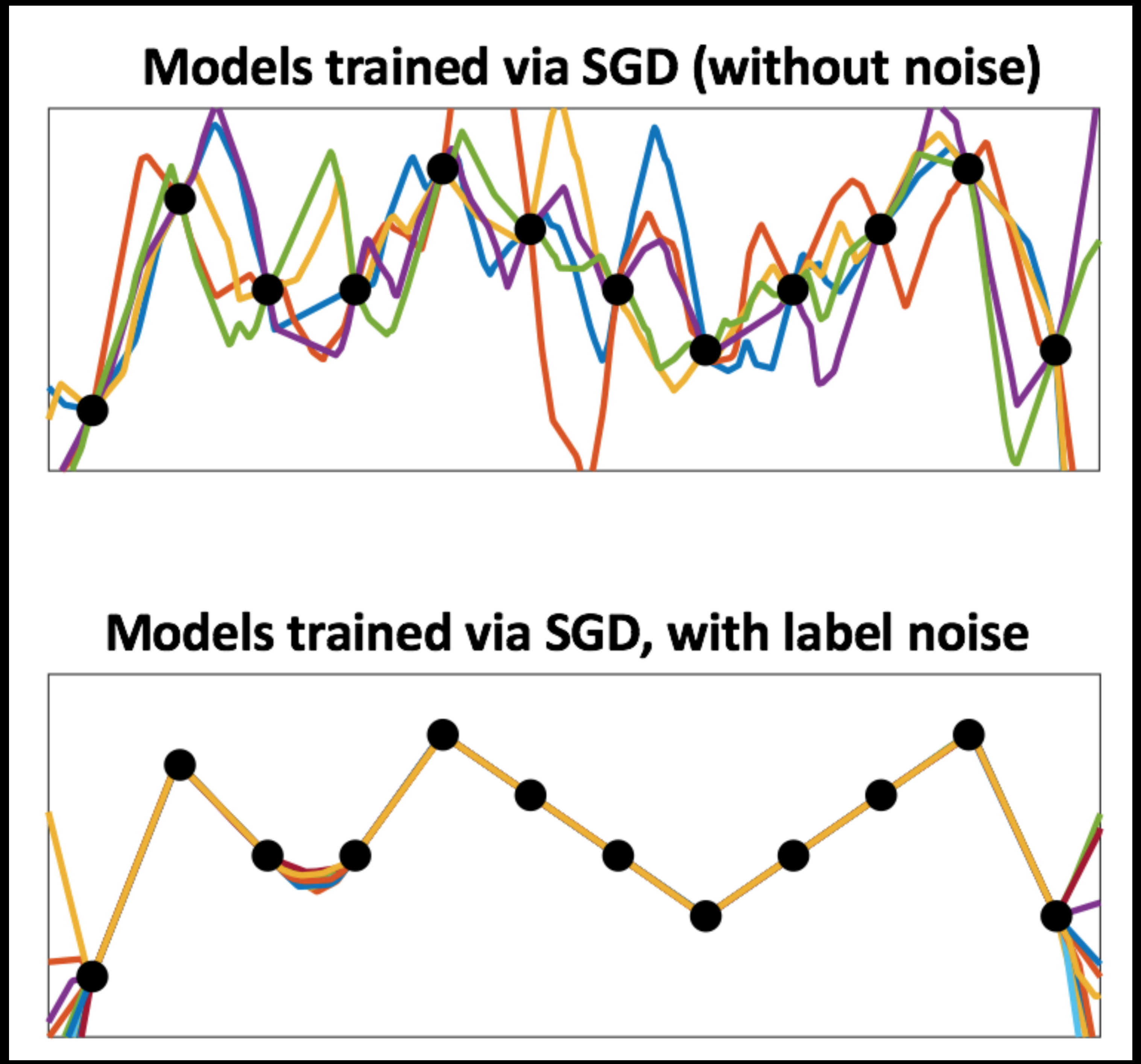


- Single hidden-layer ReLU network with input in $\mathbb{R}^d$ and parameters $\Theta = \{\theta_j = (a_j, b_j, c_j) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}\}_{j=1}^{M}$:

$$f(x;\Theta) = \frac{1}{M}\sum_{j=1}^{M} c_j (a_j^\top x + b_j)_+ \ .$$

- Goal: Memorize training set $\{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}\}_{i\leq n}$, ie find Θ^* such that $f(x_i;\Theta^*) = y_i$, with ***small*** complexity, e.g. smallest possible M, or smallest weights $\frac{1}{M}\sum_{j=1}^{M}\|\theta_j\|^2$

[Blanc et al, COLT'20]



- Questions:
  - How does gradient-descent behave under different over-parametrisation scaling and regularisation?
  - Towards optimization guarantees for finite width?

MEMORIZATION IN SHALLOW NEURAL NETWORKS: BACKGROUND

- How large should we expect M to be in order to memorize n points in dimension d?

MEMORIZATION IN SHALLOW NEURAL NETWORKS: BACKGROUND

- How large should we expect M to be in order to memorize n points in dimension d?
  - $M \geq n$ follows directly from Universal Approximation and Convex Geometry [Caratheodory]
  - In fact, $M \approx n/d$ is possible [Baum'88 for threshold units, Bubeck et al'20 for ReLU].

MEMORIZATION IN SHALLOW NEURAL NETWORKS: BACKGROUND

- How large should we expect M to be in order to memorize n points in dimension d?
  - $M \geq n$ follows directly from Universal Approximation and Convex Geometry [Caratheodory]
  - In fact, $M \approx n/d$ is possible [Baum'88 for threshold units, Bubeck et al'20 for ReLU].
  - However, number of neurons is not necessarily good notion of complexity.
  - Moreover, previous memorization algorithms do not correspond to gradient descent.

MEMORIZATION IN SHALLOW NEURAL NETWORKS: BACKGROUND

- How large should we expect M to be in order to memorize n points in dimension d?
  - $M \geq n$ follows directly from Universal Approximation and Convex Geometry [Caratheodory]
  - In fact, $M \approx n/d$ is possible [Baum'88 for threshold units, Bubeck et al'20 for ReLU].
  - However, number of neurons is not necessarily good notion of complexity.
  - Moreover, previous memorization algorithms do not correspond to gradient descent.
- Tychonov Regularisation (aka weight decay, path-norm): $\mathcal{R}(f) = \frac{1}{M}\sum_{j=1}^{M} \|\theta_j\|^2$.
  - Sparsity $\widetilde{O}(n/d)$ with total weight $\mathcal{R}(f) = \widetilde{O}(\sqrt{n})$ sufficient [Bubeck et al], but not gradient-descent.

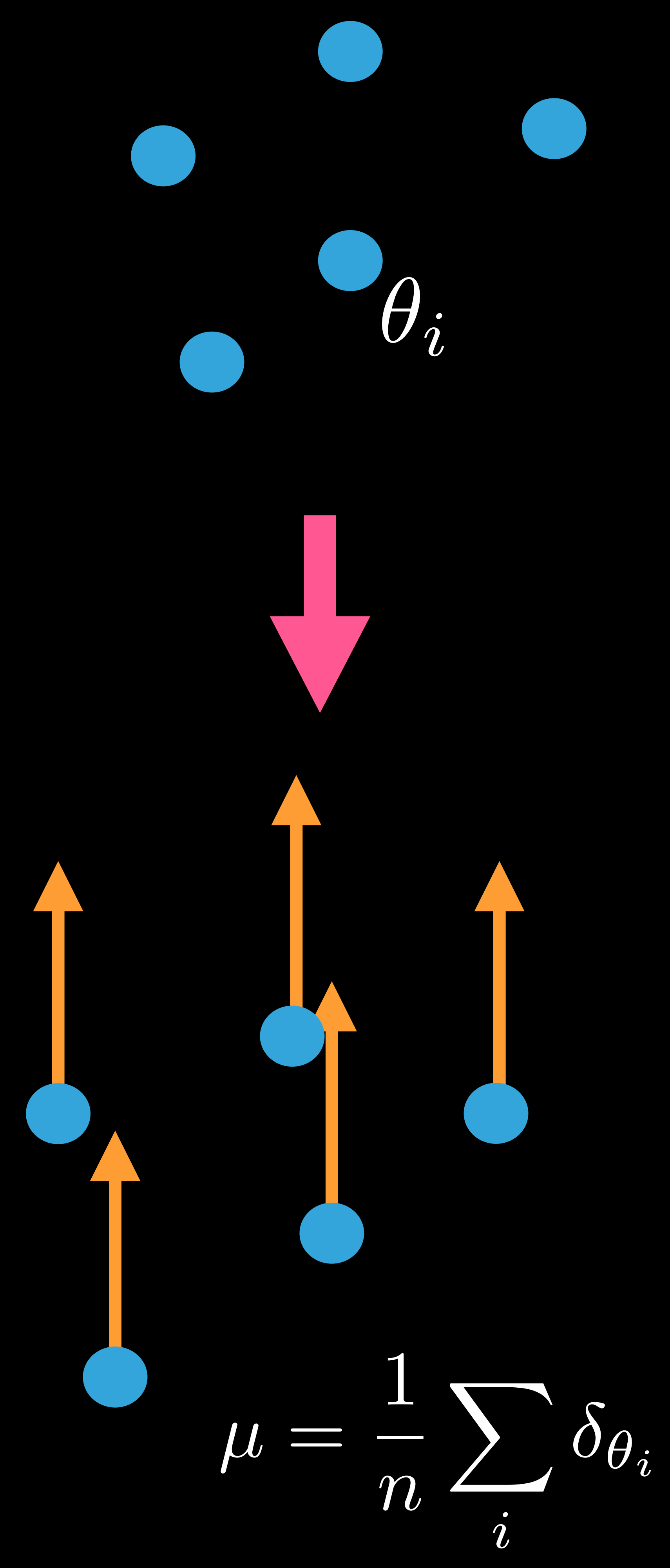
MEMORIZATION IN SHALLOW NEURAL NETWORKS: BACKGROUND

- How large should we expect M to be in order to memorize n points in dimension d?
  - $M \geq n$ follows directly from Universal Approximation and Convex Geometry [Caratheodory]
  - In fact, $M \approx n/d$ is possible [Baum'88 for threshold units, Bubeck et al'20 for ReLU].
  - However, number of neurons is not necessarily good notion of complexity.
  - Moreover, previous memorization algorithms do not correspond to gradient descent.
- Tychonov Regularisation (aka weight decay, path-norm): $\mathcal{R}(f) = \frac{1}{M}\sum_{j=1}^{M} \|\theta_j\|^2$.
  - Sparsity $\widetilde{O}(n/d)$ with total weight $\mathcal{R}(f) = \widetilde{O}(\sqrt{n})$ sufficient [Bubeck et al], but not gradient-descent.
- Gradient Descent analysis in the random feature (=kernel) regime
  - [Daniely'20] shows $\widetilde{O}(n/d)$ are sufficient, but poor generalisation.
  - How about active, non-linear regime?

LIFTING TO MEASURES OVER PARAMETERS

- For each choice of parameters $\Theta = \{\theta_j = (a_j, b_j, c_j) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}\}_{j=1}^{M}$ we can associate an empirical measure $\hat{\mu} = \frac{1}{M}\sum_{j=1}^{M} \delta_{\theta_j}$ defined in $\Omega = \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}$, so that

$$f(x;\Theta) = \int_\Omega c(a^\top x + b)_+ d\mu(a,b,c)$$



[Rosset et al, Bengio et al, Bach]
[Mei et al, Chizat et al]
[Rotskoff et al, Sirignano et al]

LIFTING TO MEASURES OVER PARAMETERS

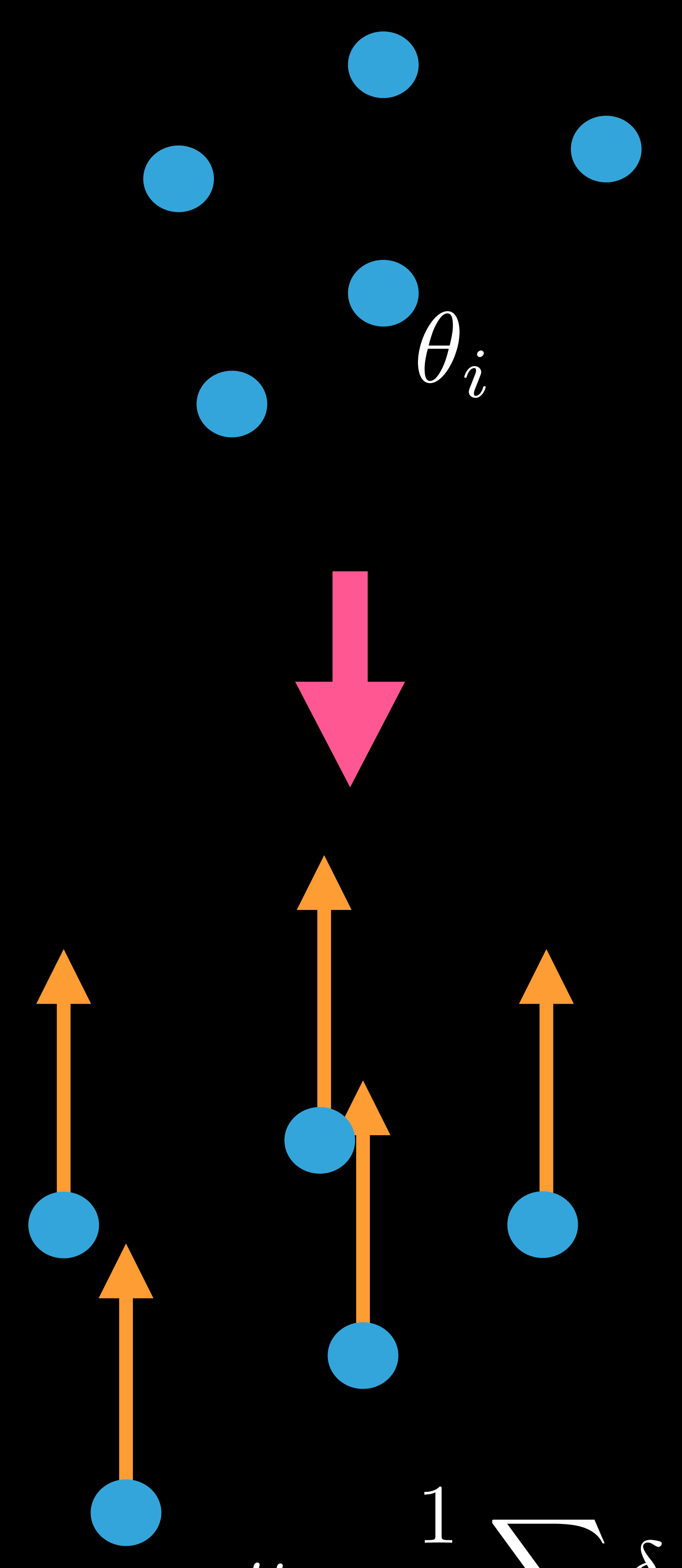
- For each choice of parameters $\Theta = \{\theta_j = (a_j, b_j, c_j) \in \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}\}_{j=1}^{M}$ we can associate an empirical measure $\hat{\mu} = \frac{1}{M}\sum_{j=1}^{M} \delta_{\theta_j}$ defined in $\Omega = \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}$, so that

$$f(x;\Theta) = \int_\Omega c(a^\top x + b)_+ d\mu(a,b,c)$$

- Tychonov-Regularised Memorization problem becomes

$$\min_\mu \int_\Omega \|\theta\|^2 d\mu(\theta) \quad \text{s.t. } f(x_i;\mu) = y_i, i \in [n].$$

  - From the Representer Theorem, sparse solution exists with at most n atoms.
  - Similar geometry using implicit regularisation with label noise [Blanc et al.'20]
  - Structure of general solutions?



$$\mu = \frac{1}{n}\sum_i \delta_{\theta_i}$$

[Rosset et al, Bengio et al, Bach]
[Mei et al, Chizat et al]
[Rotskoff et al, Sirignano et al]

BACK TO FINITE-DIMENSIONAL LINEAR PROGRAM

- Overparametrised memorization "hides" an underlying finite-dimensional linear program:

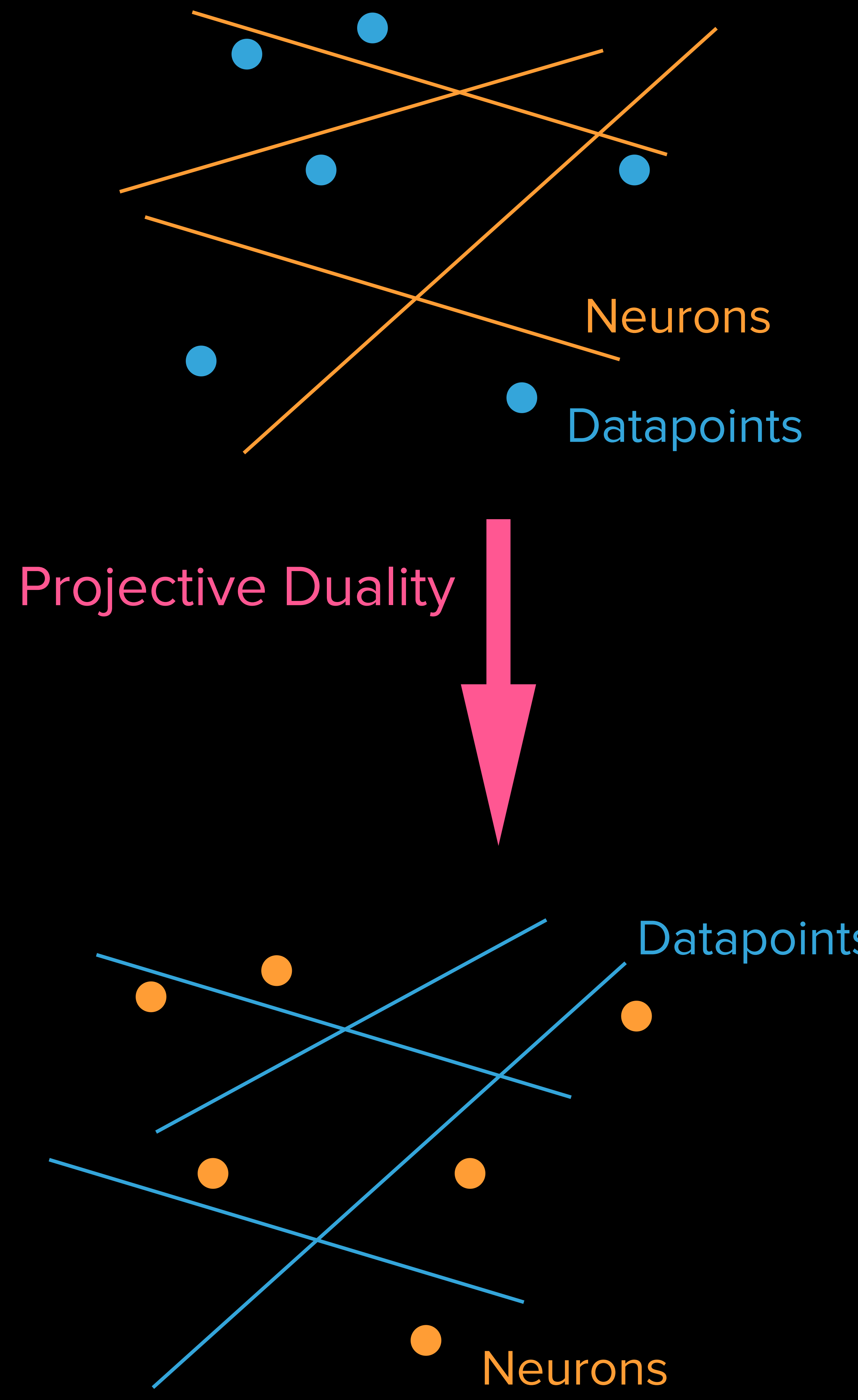
Theorem: [DB'20] Any minimiser μ^* of the ReLU Tychonov memorization problem has atomic support of at most $O(n)^{O(d)}$ points (after removing the symmetries in the parametrisation).

BACK TO FINITE-DIMENSIONAL LINEAR PROGRAM

- Overparametrised memorization "hides" an underlying finite-dimensional linear program:

Theorem: **[DB'20]** Any minimiser μ^* of the ReLU Tychonov memorization problem has atomic support of at most $O(n)^{O(d)}$ points (after removing the symmetries in the parametrisation).

- What is the nature of this linear program?
  - Each datapoint defines a hyperplane in $\Omega \cong \mathbb{R}^{d+2}$.
  - n datapoints define a hyperplane arrangement in Ω with $S = O(n)^{O(d)}$ cells.
  - μ^* necessarily concentrates in at most one point $\bar{\theta}_s$ for each cell.



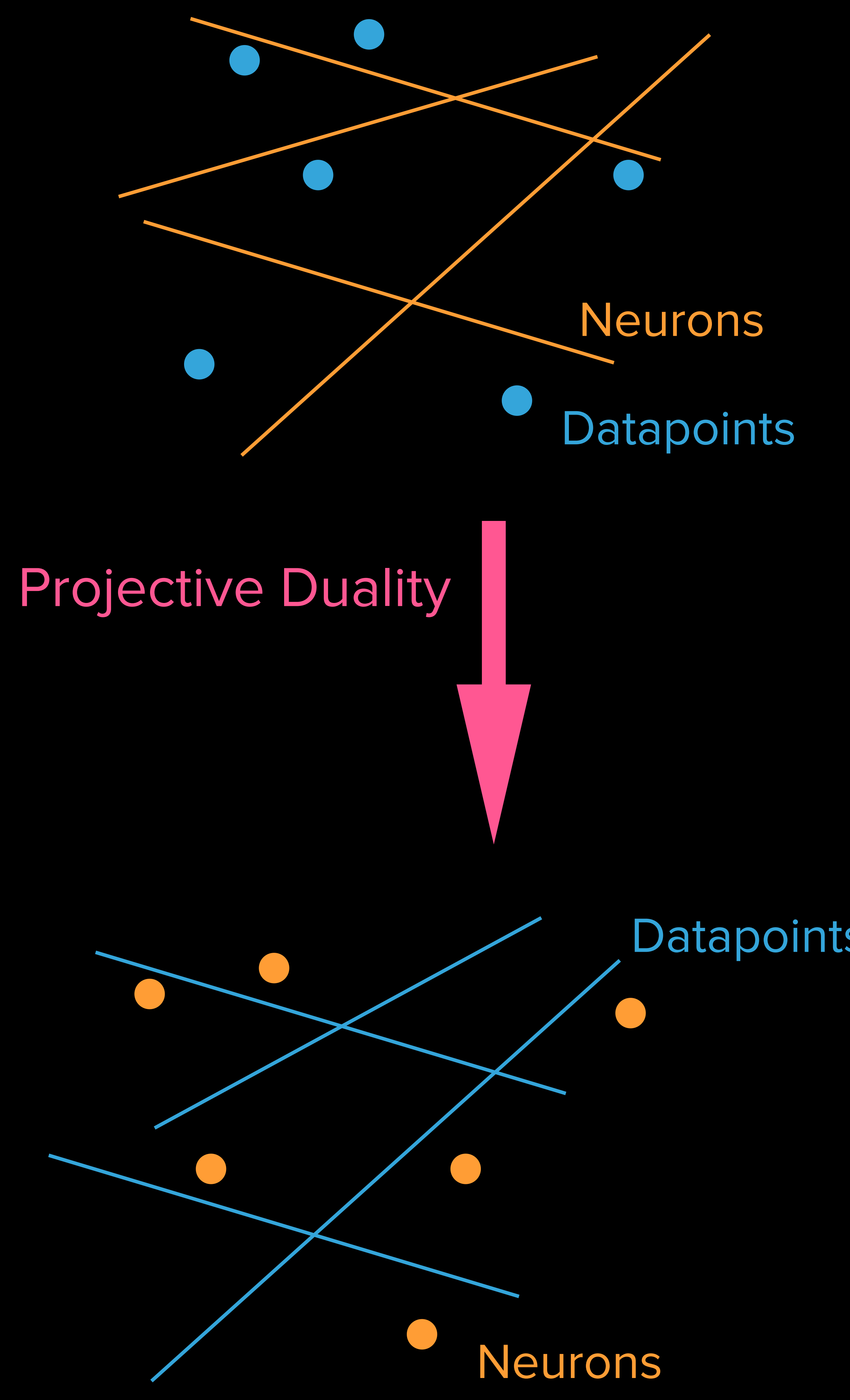
BACK TO FINITE-DIMENSIONAL LINEAR PROGRAM

- Overparametrised memorization "hides" an underlying finite-dimensional linear program:

Theorem: [DB'20] Any minimiser μ^* of the ReLU Tychonov memorization problem has atomic support of at most $O(n)^{O(d)}$ points (after removing the symmetries in the parametrisation).

- What is the nature of this linear program?
  - Each datapoint defines a hyperplane in $\Omega \cong \mathbb{R}^{d+2}$.
  - n datapoints define a hyperplane arrangement in Ω with $S = O(n)^{O(d)}$ cells.
  - μ^* necessarily concentrates in at most one point $\bar{\theta}_s$ for each cell.
- As a result, minimisers $\mu^* = \sum_{s=1}^{S} z_s \delta_{\bar{\theta}_s}$ are solutions of

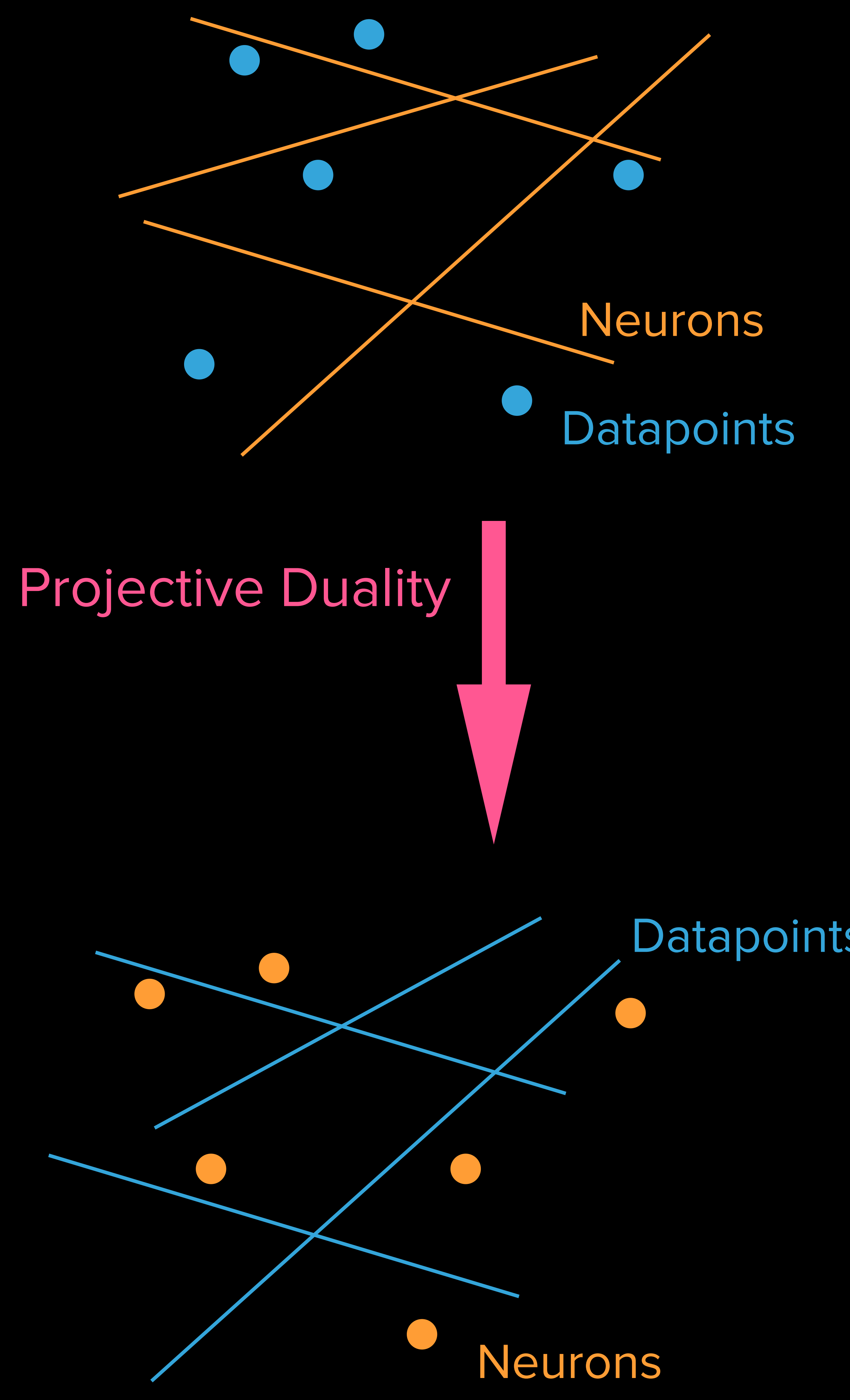
$$\min \|z\|_1 \quad \text{s.t. } \mathcal{A}z = y \quad \text{with} \quad \mathcal{A} \in \mathbb{R}^{n \times S},\ \mathcal{A}_{i,s} = \langle x_i, \bar{\theta}_s \rangle_+$$



CURRENT AND FUTURE QUESTIONS

$$\min \|z\|_1 \quad \text{s.t. } \mathcal{A}z = y \quad \text{with} \quad \mathcal{A} \in \mathbb{R}^{n \times S},\ \mathcal{A}_{i,s} = \langle x_i, \bar{\theta}_s \rangle_+$$

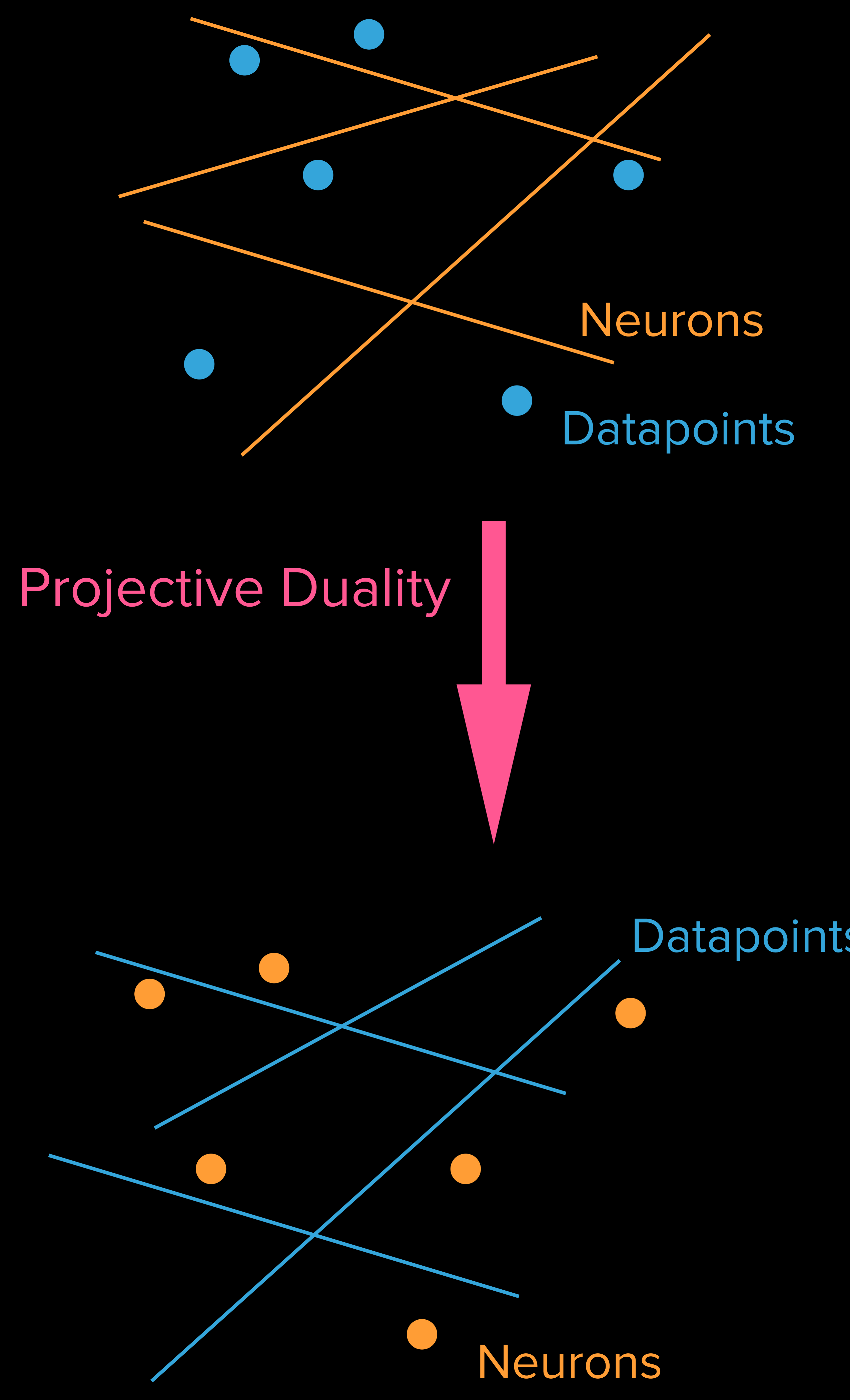
- The sensing matrix $\mathcal{A}$ is highly coherent/redundant ($S \gg n$)
- We know a solution exists with support at most n. (Representer theorem)
  - Open: RIP at level $\text{poly}(d, n)$?



CURRENT AND FUTURE QUESTIONS

$$\min \|z\|_1 \quad \text{s.t. } \mathcal{A}z = y \quad \text{with} \quad \mathcal{A} \in \mathbb{R}^{n \times S}, \mathcal{A}_{i,s} = \langle x_i, \bar{\theta}_s \rangle_+$$

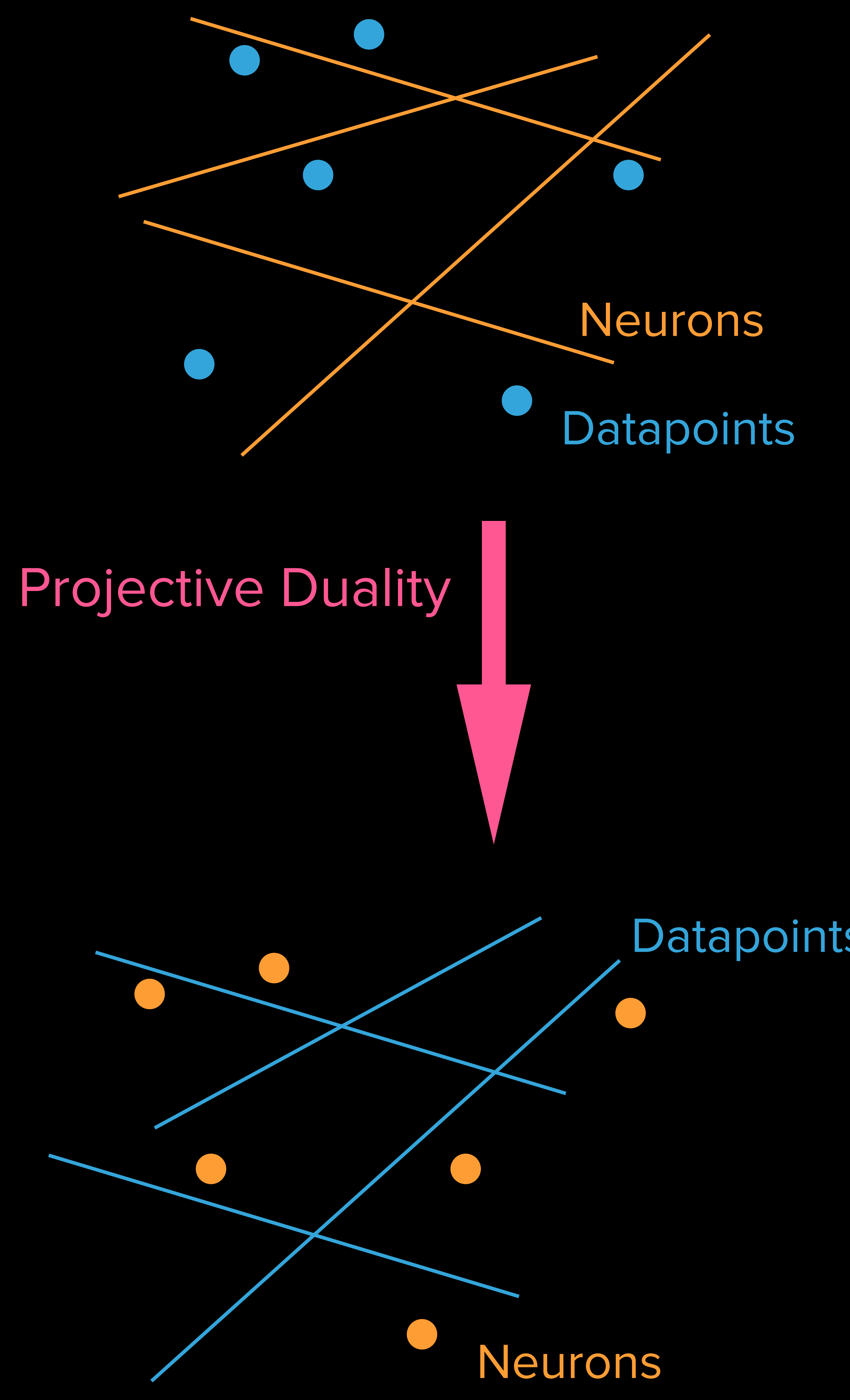
- The sensing matrix $\mathcal{A}$ is highly coherent/redundant ($S \gg n$)
- We know a solution exists with support at most n. (Representer theorem)
  - Open: RIP at level poly(d, n) ?
- Towards gradient Descent Guarantees for finite width:
  - We have local curvature of the loss in the measure space [Chizat'19, Ge, Jin'21]
  - Main technical challenge: lack of smoothness of the training map.



CURRENT AND FUTURE QUESTIONS

$$\min \|z\|_1 \quad \text{s.t. } \mathcal{A}z = y \quad \text{with} \quad \mathcal{A} \in \mathbb{R}^{n\times S},\ \mathcal{A}_{i,s} = \langle x_i, \bar{\theta}_s \rangle_+$$

- The sensing matrix $\mathcal{A}$ is highly coherent/redundant ($S \gg n$)
- We know a solution exists with support at most n. (Representer theorem)
  - Open: RIP at level poly(d, n) ?
- Towards gradient Descent Guarantees for finite width:
  - We have local curvature of the loss in the measure space [Chizat'19, Ge, Jin'21]
  - Main technical challenge: lack of smoothness of the training map.
  - Current/Open: leverage piece-wise smoothness of the map.
  - Average-vs-worst case rates (SQ-lower bounds) [Goel et al, Diak.]



FUNCTION APPROXIMATION OF SPARSE INFERENCE

- Recall sparse inference task: given dictionary $W \in \mathbb{R}^{d \times m}$, $m > d$, and $x = Wz$, recover z by exploiting a sparsity prior. $f^*_W(x) := \arg\min \{\|z\|_0;\ x = Wz\}$.

- Main algorithmic paradigm: relax ℓ_0 to ℓ_1 and consider the penalized quadratic program

$$\text{Lasso}\quad \tilde{f}_W(x) := \arg\min_z \left\{\|x - Wz\|^2 + \lambda\|z\|_1\right\} .$$

[Tibshirani]

  - Solved e.g using Iterative Soft-Thresholding Algorithm (ISTA, Proximal Gradient descent).

FUNCTION APPROXIMATION OF SPARSE INFERENCE

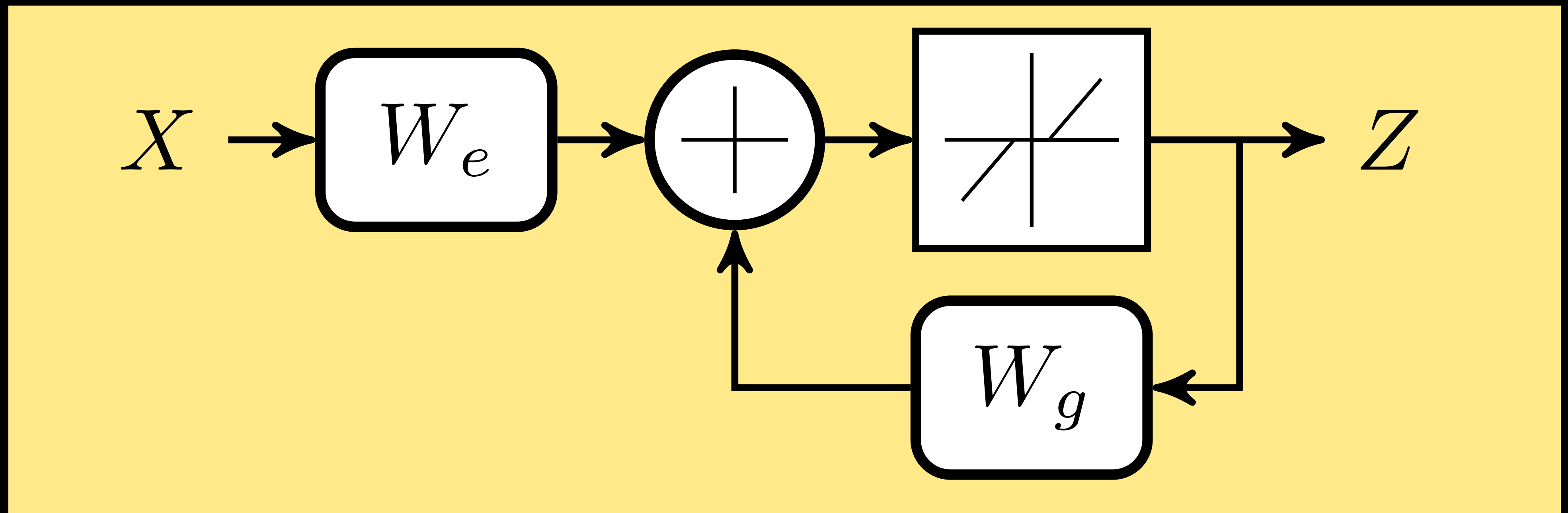
- Recall sparse inference task: given dictionary $W \in \mathbb{R}^{d \times m}$, $m > d$, and $x = Wz$, recover z by exploiting a sparsity prior. $f^*_W(x) := \arg\min \{\|z\|_0;\ x = Wz\}$.

- Main algorithmic paradigm: relax ℓ_0 to ℓ_1 and consider the penalized quadratic program

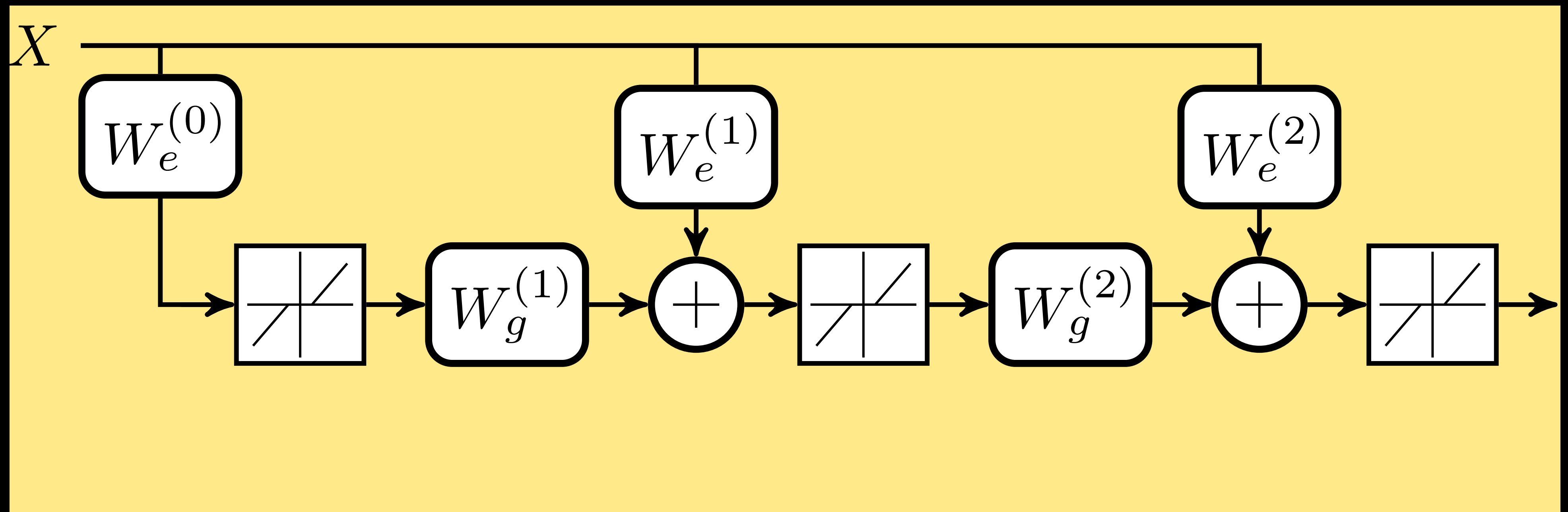
$$\text{Lasso} \quad \tilde{f}_W(x) := \arg\min_z \left\{ \|x - Wz\|^2 + \lambda \|z\|_1 \right\} .$$ [Tibshirani]

  - Solved e.g using Iterative Soft-Thresholding Algorithm (ISTA, Proximal Gradient descent).

- By unrolling this iterative scheme, [Gregor & LeCun] propose a neural network approximation, LISTA:



ISTA — Recurrent Neural Network



LISTA — Unfolded Neural Network

FUNCTION APPROXIMATION OF SPARSE INFERENCE

- Recall sparse inference task: given dictionary $W \in \mathbb{R}^{d \times m}$, $m > d$, and $x = Wz$, recover z by exploiting a sparsity prior. $f^*_W(x) := \arg\min \{\|z\|_0;\ x = Wz\}$.

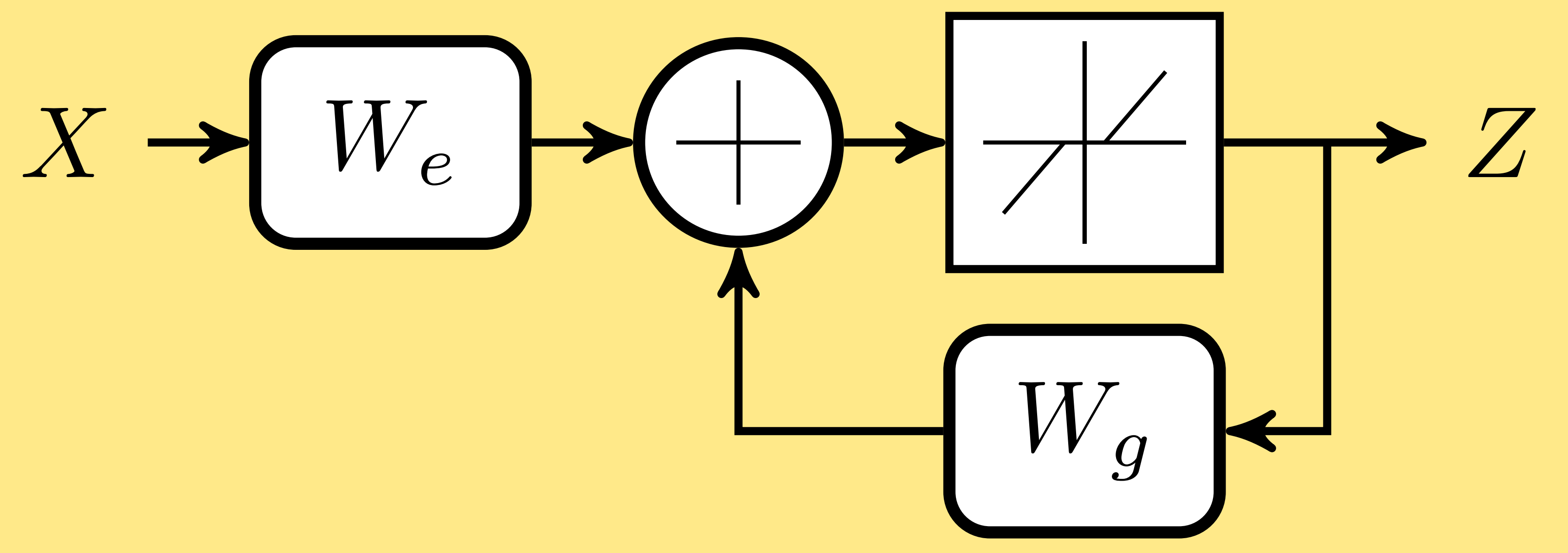
- Main algorithmic paradigm: relax ℓ_0 to ℓ_1 and consider the penalized quadratic program

$$\text{Lasso} \quad \tilde{f}_W(x) := \arg\min_z \left\{ \|x - Wz\|^2 + \lambda \|z\|_1 \right\} .$$

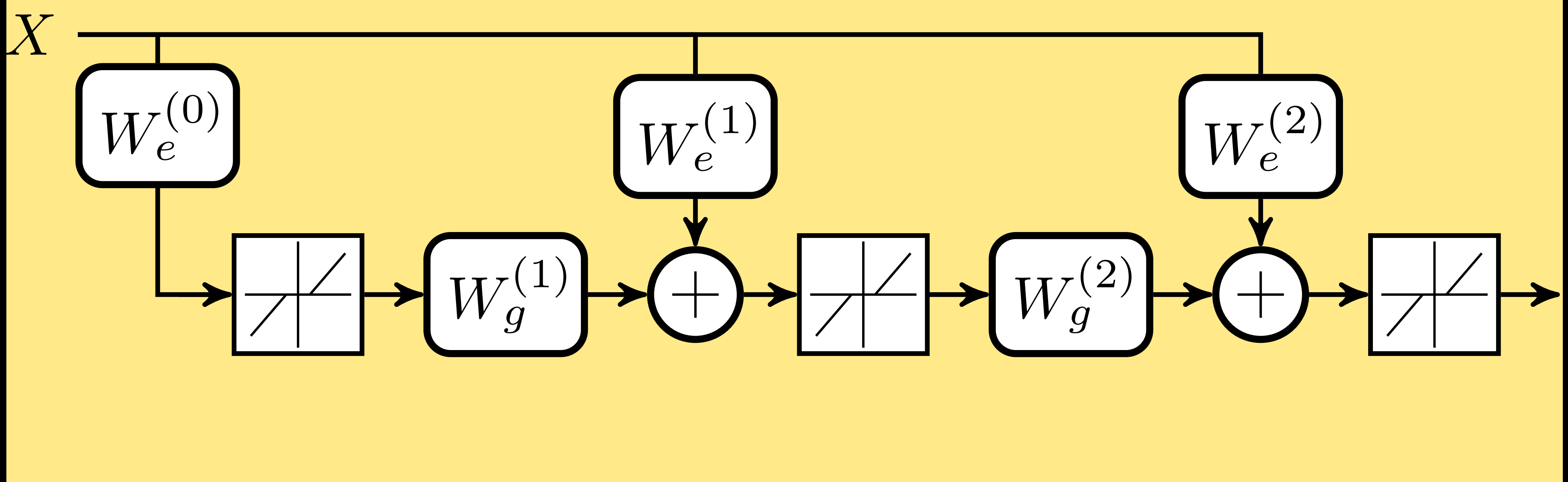
[Tibshirani]

  - Solved e.g using Iterative Soft-Thresholding Algorithm (ISTA, Proximal Gradient descent).

- By unrolling this iterative scheme, [Gregor & LeCun] propose a neural network approximation, LISTA:



ISTA — Recurrent Neural Network

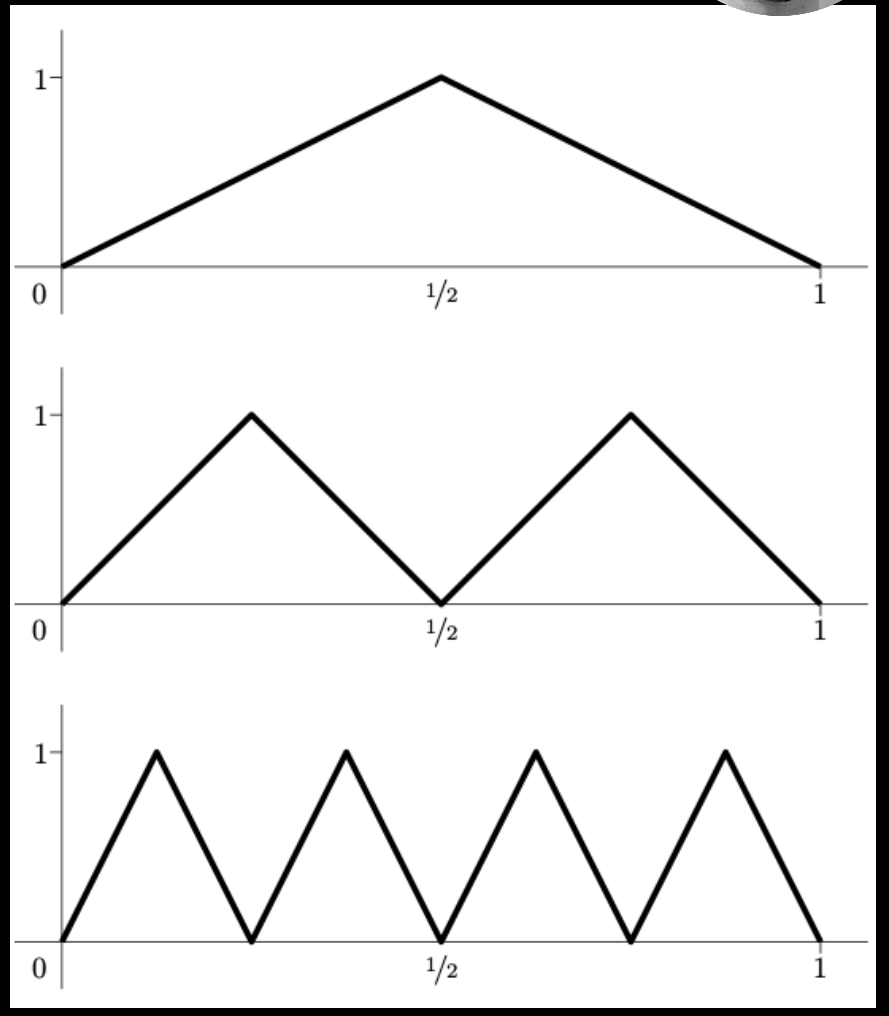


LISTA — Unfolded Neural Network

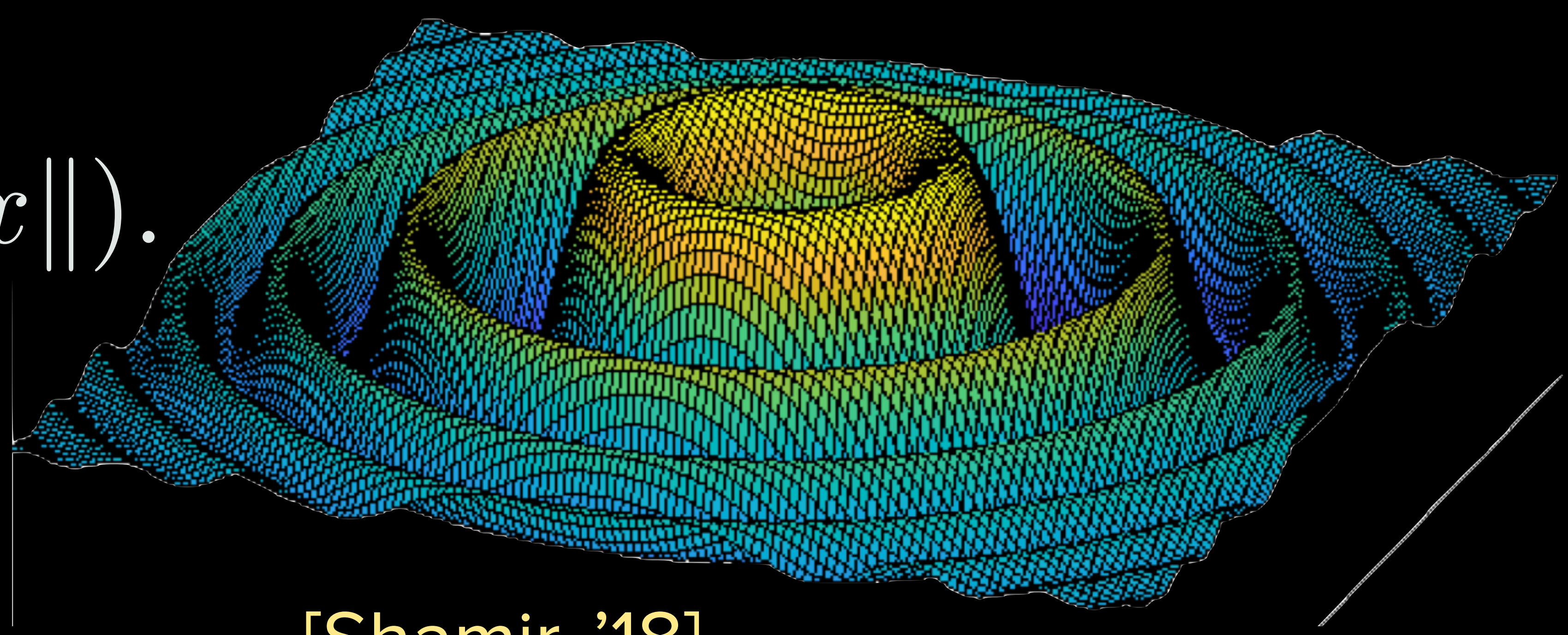
- Unrolling iterative algorithm is sufficient. Is it also necessary?
- Depth-width tradeoffs for such sparse inference?

DEPTH SEPARATION PRIOR WORK

- Rich literature in boolean [Rossman, Hastad'68] or threshold [Hajnal'93] circuit lower bounds.
- [Martens et al'13] shows lower bounds for RBMs.
- [Telgarsky'15] Exploits combinatorial limitations of shallow networks
  - Refined periodicity analysis in [Chatziafratis et al'20].
- [Montufar et al.] bound number of linear regions of deep ReLU nets.
- [Eldan, Shamir, Safran, Daniely] construct oscillatory functions with depth-separation. Provably require $\exp(d)$ width for shallow model, but $\text{poly}(d)$ for deeper neural network.
  - Constructions are inherently low-dimensional, e.g. $f(x) = g(\|x\|)$.
- Depth Separation for sparse inference?



[Telgarsky, '15]



[Shamir, '18]

DEPTH SEPARATION BEYOND RADIAL FUNCTIONS

- Key ingredients for depth separation: functions with oscillatory behavior and heavy-tailed input data distributions:

Theorem [BJV'20]: Let $f^*(x) = \exp\{i\langle \omega_d, \rho(Ux + b)\rangle\}$ with $U \in \mathbb{R}^{d\times d}$, $\|\omega_d\| = \Omega(d^3)$ and $\rho(t) = \max(0, t)$. Let μ be a heavy-tailed distribution. Then

(i) f^* is not $\Omega(1)$-approximable by any shallow $\mathsf{exp}(o(d))$-wide network.

(ii) there exists a $\mathsf{poly}(d, \epsilon^{-1})$ 3-layer ReLU network f such that $D_\mu(f, f^*) \leq \epsilon$.

$$D_\mu(f, g) = \mathbb{E}_\mu |f(x) - g(x)|^2$$

DEPTH SEPARATION BEYOND RADIAL FUNCTIONS


- Key ingredients for depth separation: functions with oscillatory behavior and heavy-tailed input data distributions:

Theorem [BJV'20]: Let $f^*(x) = \exp\{i\langle \omega_d, \rho(Ux+b)\rangle\}$ with $U \in \mathbb{R}^{d\times d}$, $\|\omega_d\| = \Omega(d^3)$ and $\rho(t) = \max(0,t)$. Let μ be a heavy-tailed distribution. Then

(i) f^* is not $\Omega(1)$-approximable by any shallow $\mathsf{exp}(o(d))$-wide network.

(ii) there exists a $\mathsf{poly}(d, \epsilon^{-1})$ 3-layer ReLU network f such that $D_\mu(f, f^*) \leq \epsilon$.

$$D_\mu(f,g) = \mathbb{E}_\mu |f(x) - g(x)|^2$$

- Deep Piece-wise linear functions over compact domains are easier to approximate with shallow models:

Theorem [BJV'20]: Let $f^*(x)$ be a depth-L ReLU network with weights $\|W_l\|_\infty = \Theta(1)$ for $l \leq L$. Then $\forall \epsilon > 0$ there is a shallow ReLU network f_n such that $D_{\mathbb{S}^d,\infty}(f^*, f_n) \leq \epsilon$ of width

$$n \geq \left(\Theta(\exp L)(1+\epsilon^{-2})\mathsf{poly}(d)\right)^{\Omega(\epsilon^{-L})}.$$

  - Extends previous results in [Safran, Eldan, Shamir'19] for radial functions.
  - Rate is polynomial in d, but exponential in ϵ^{-1}.

APPLICATION TO SPARSE INFERENCE

- Since ISTA iterations are piece-wise linear, we can leverage this upper bound for sufficiently incoherent dictionaries:

> **Corollary [VB'21]:** Let $m = \rho d$, $k = \alpha d$ with $\rho > 1, \alpha < 1$. Let ν_d be the uniform measure over k-sparse unit-norm m-dimensional vectors, and assume $W \in \mathbb{R}^{d \times m}$ satisfies RIP $\delta_{2k}(W) \leq 0.6$. For each $\epsilon > 0$, there exists a shallow network f_M such that $D_{\nu_d}(f^*_W, f_M) \leq \epsilon$ of width $\mathsf{poly}(d)$.

- Rate is polynomial in d, but exponential in ϵ^{-1}.
- Depth can still provide substantial improvements in approximation.
- Data adaptivity: rates may be improved by localizing.

APPLICATION TO SPARSE INFERENCE

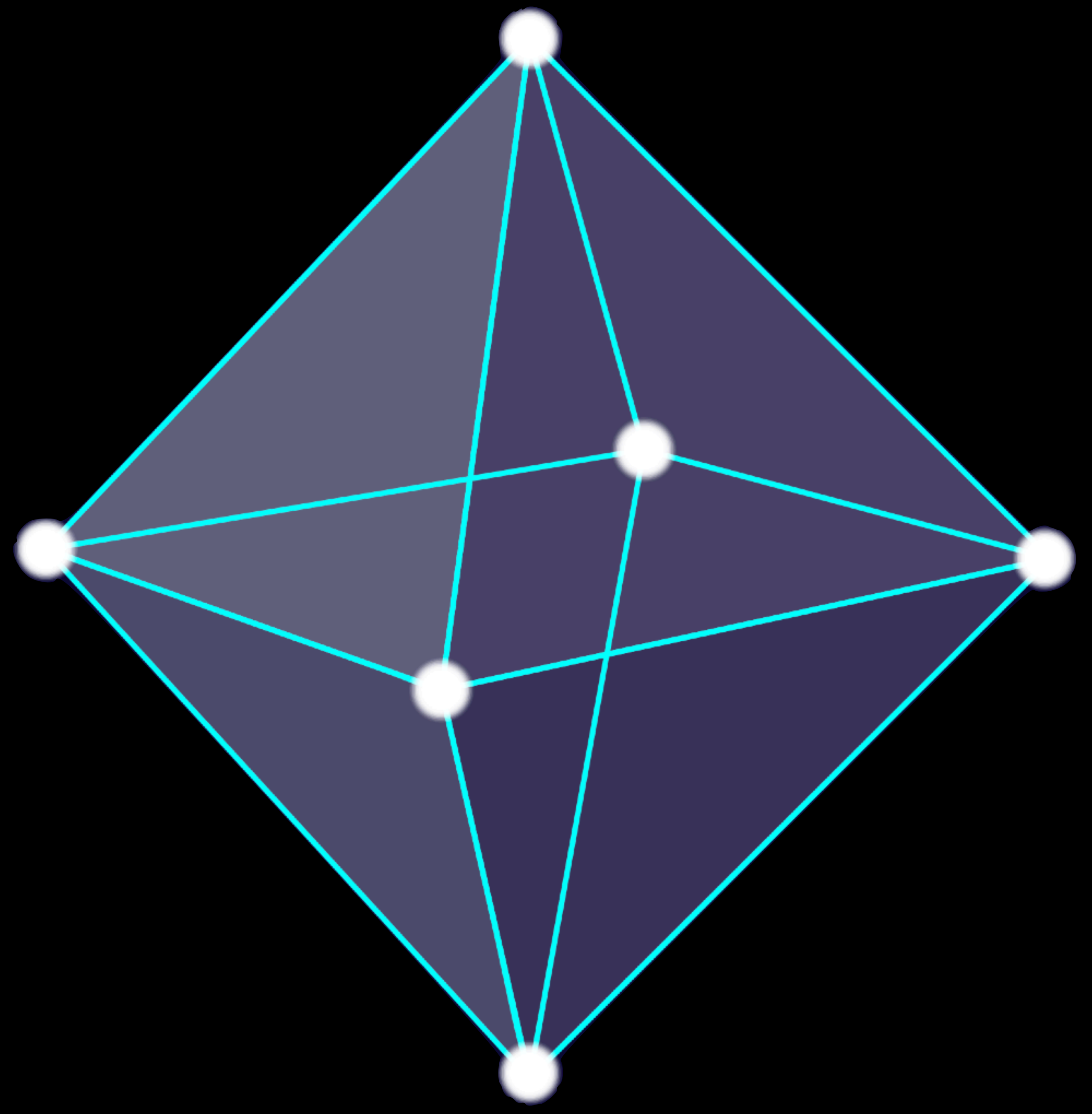
- Since ISTA iterations are piece-wise linear, we can leverage this upper bound for sufficiently incoherent dictionaries:

Corollary [VB'21]: Let $m = \rho d$, $k = \alpha d$ with $\rho > 1, \alpha < 1$. Let ν_d be the uniform measure over k-sparse unit-norm m-dimensional vectors, and assume $W \in \mathbb{R}^{d \times m}$ satisfies RIP $\delta_{2k}(W) \leq 0.6$. For each $\epsilon > 0$, there exists a shallow network f_M such that $D_{\nu_d}(f^*_W, f_M) \leq \epsilon$ of width $\mathsf{poly}(d)$.

  - Rate is polynomial in d, but exponential in ϵ^{-1}.
  - Depth can still provide substantial improvements in approximation.
  - Data adaptivity: rates may be improved by localizing.
- Current: formalize lower bound in weaker sparsity / coherent assumptions.
- Open: optimization guarantees of learnt sparse coding.
- Open: refined analysis under more stringent sparsity conditions [Liu et al]

TAKE-HOME

- Sparse regression: rich CO problem where data geometry enables efficient algorithms.
- Sparse regression in data memorization using overparametrised shallow models:
  - Important tool to establish generic efficient learnability.
  - Geometry of hyperplane arrangement sensing matrices.



- Function Approximation of Sparse Regression
  - Shallow neural approximation not cursed by dimension.
  - Which inverse problems provably require depth? Learnability guarantees?
  - Towards structured problems (eg in graphs, grids).

THANKS!

References:

"Depth Separation beyond Radial Functions", Bruna, Jelassi, Ozuch Venturi, https://arxiv.org/abs/2102.01621v2 *preprint* 2021

"On Sparsity for Overparametrised ReLU Networks", Jaume de Dios, Bruna, https://arxiv.org/abs/2006.10225 *preprint* 2020.