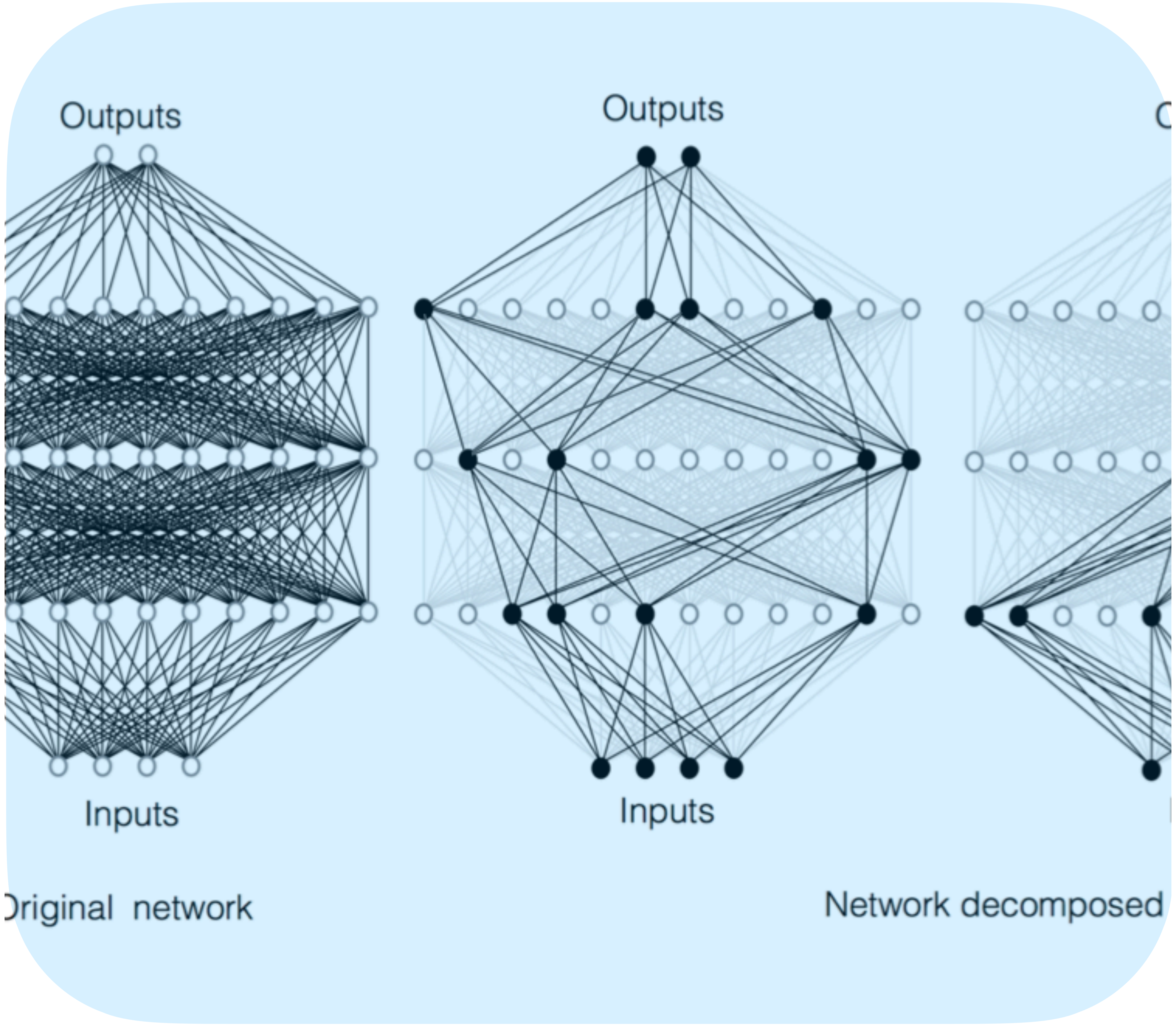
Leveraging Sparse Input and Sparse Models: Efficient Distributed Learning in Resource-Constrained Environments

Emmanouil Kariotakis[1*], Grigorios Tsagkatakis[2,3], Panagiotis Tsakalides[2,3], **Anastasios Kyrillidis[4]**
[1]KU Leuven, [2]Institute of CS - FORTH, [3]University of Crete (CS), [4]Rice University CS

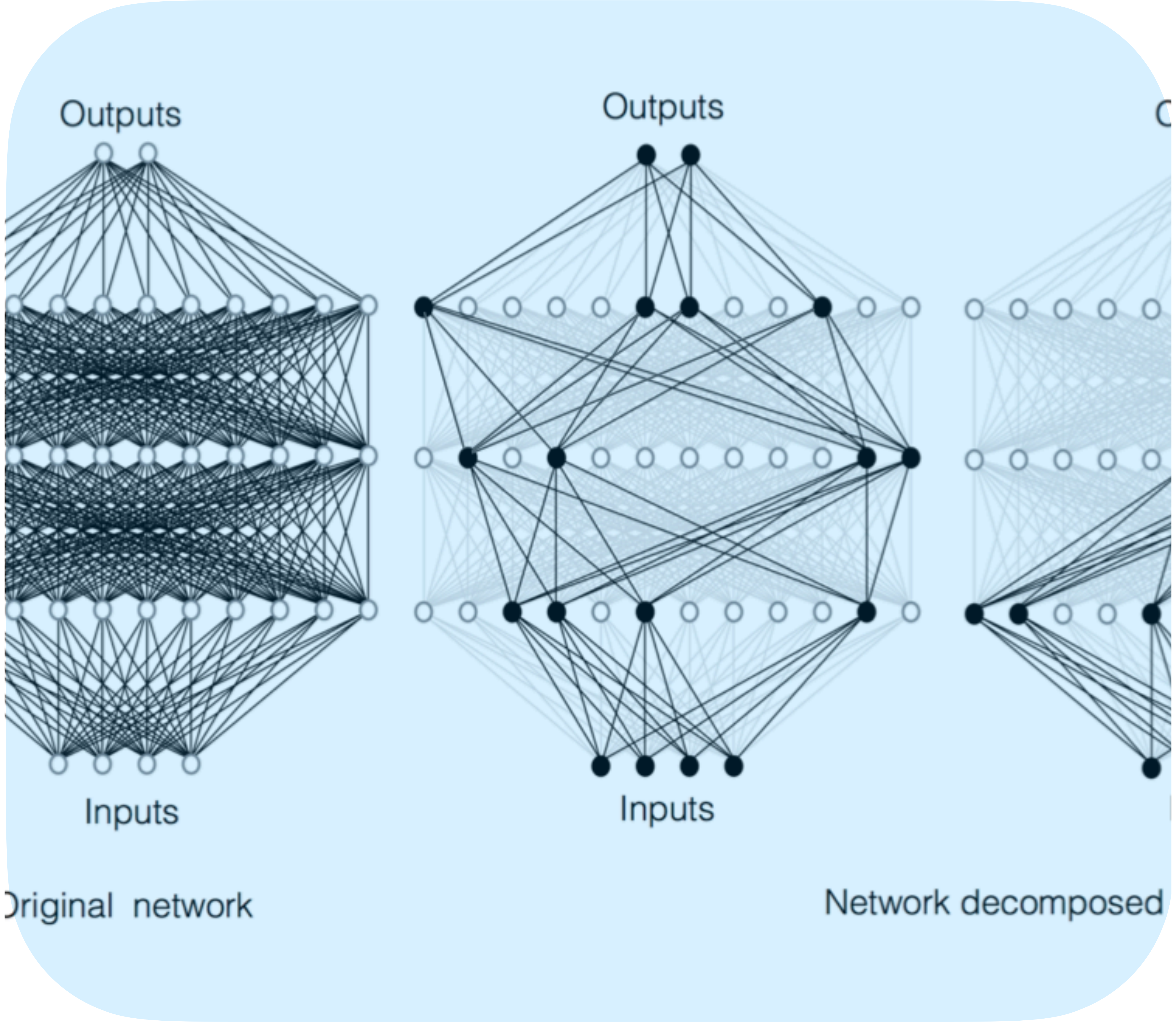
Conference on Parsimony and Learning (CPAL)

Presented by: Daniel LeJeune (Stanford)
Thank you, Daniel!

Motivation for this work: End-2-end sparse training

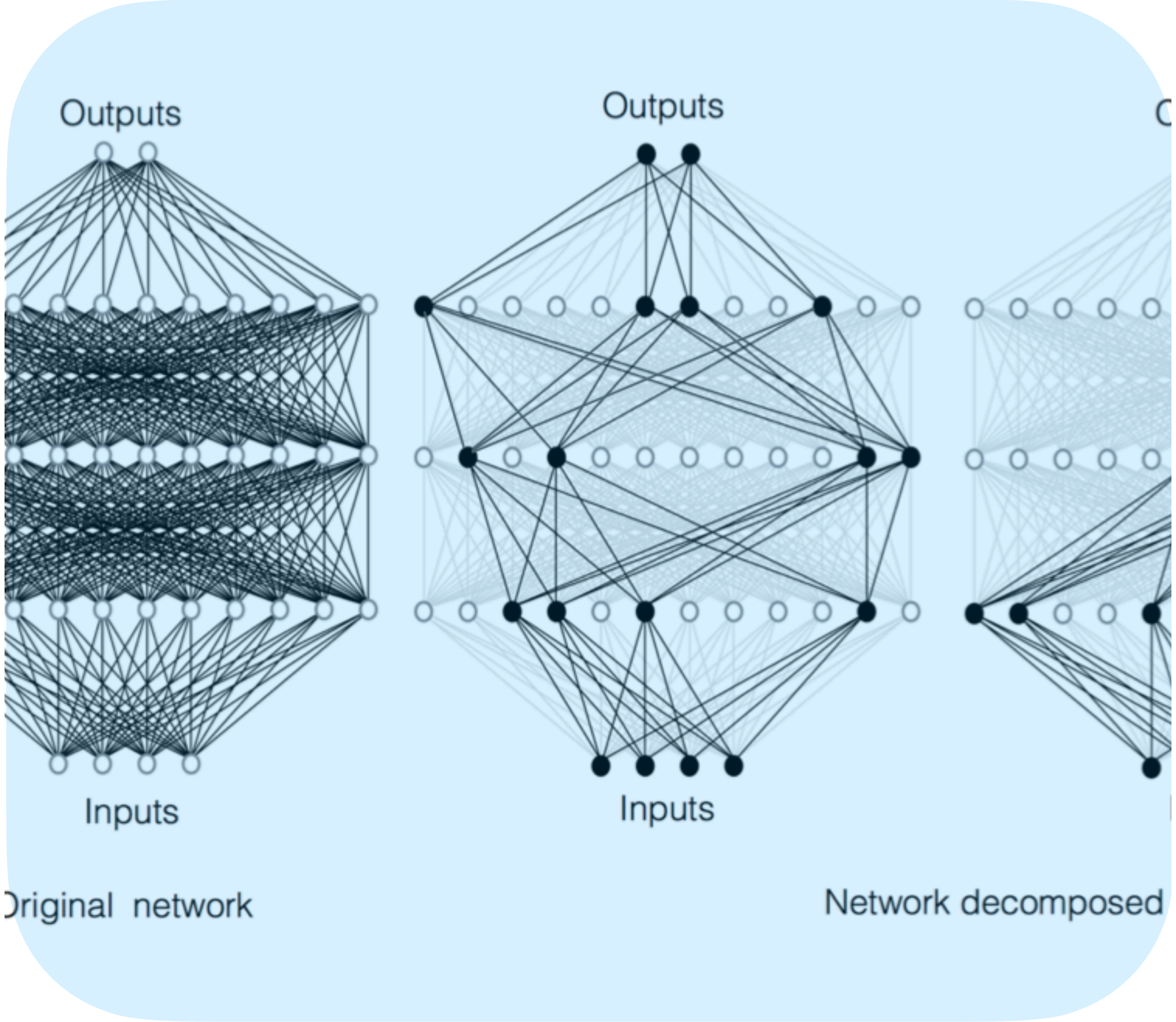


Motivation for this work: End-2-end sparse training



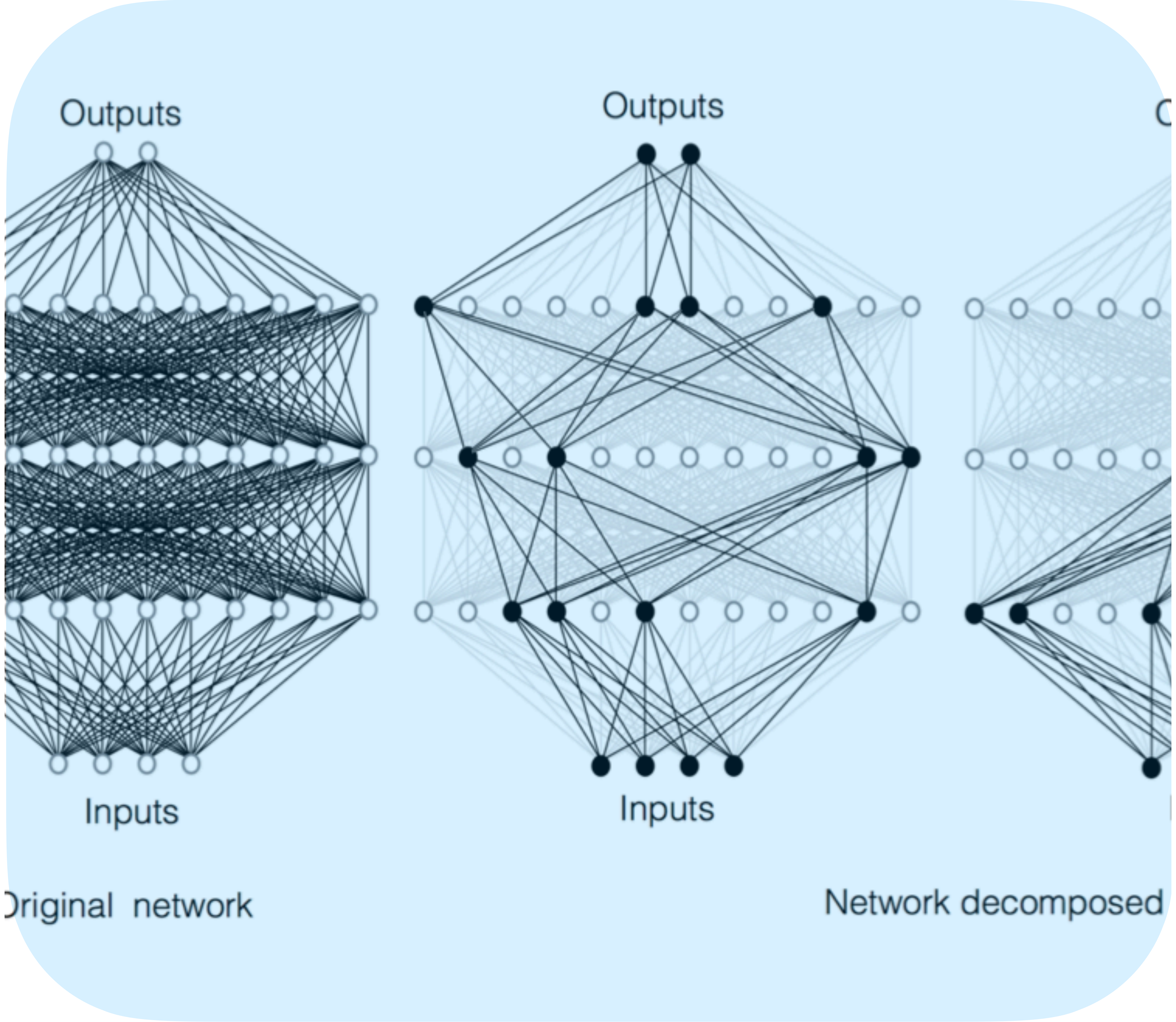
- This work is part of a bigger “journey”

Motivation for this work: End-2-end sparse training



- This work is part of a bigger "journey"
- *"Is it possible to train big models by training smaller versions of them?"*

Motivation for this work: End-2-end sparse training



- This work is part of a bigger "journey"
- *"Is it possible to train big models by training smaller versions of them?"*
- Past work answered this question affirmatively:

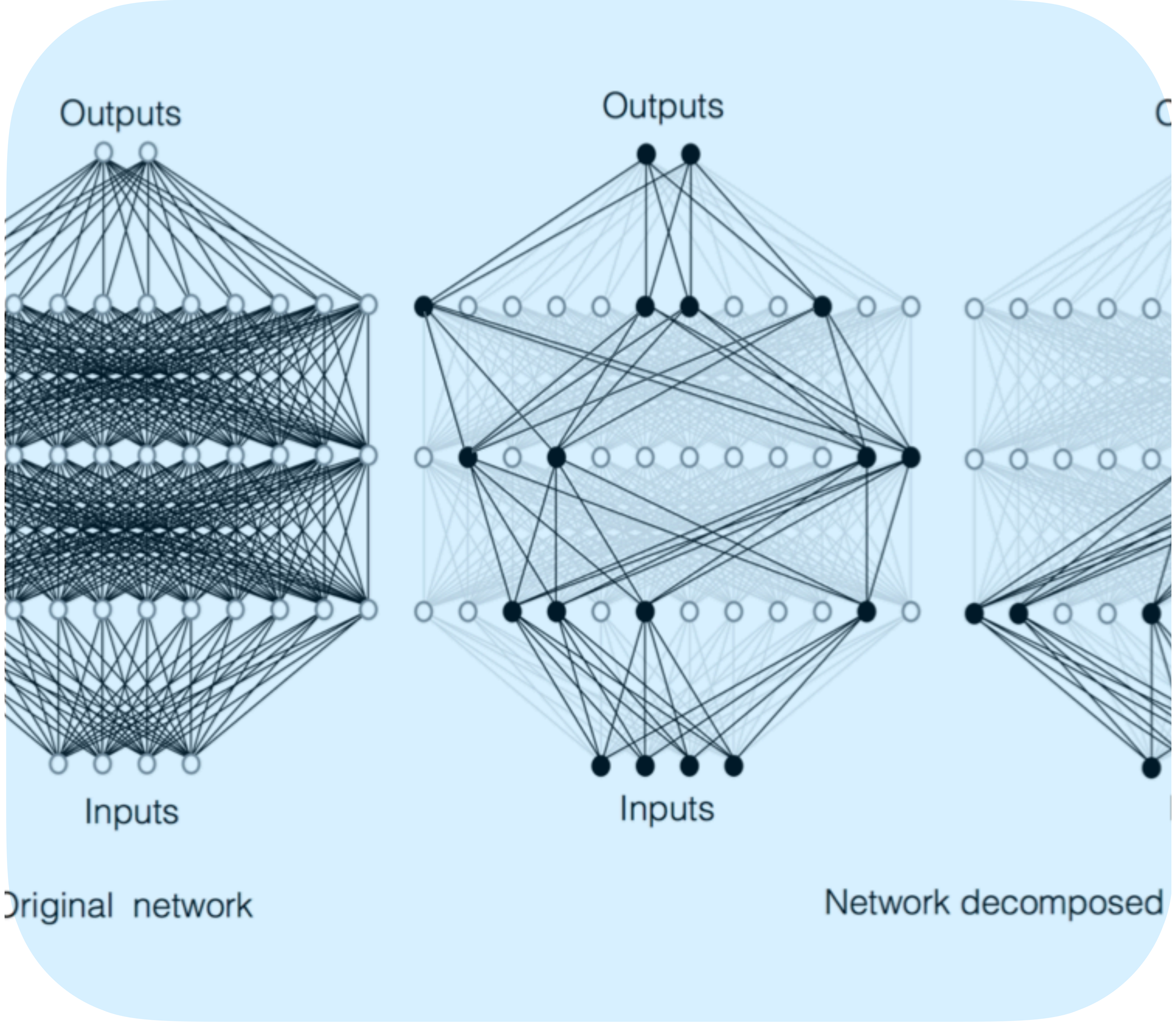
  Advertising own work: IST

  [Yuan et al. 2022, Liao et al 2021, Dun et al., 2022-23; Wan et al. 2022, Hu et al., 2023, Wang et al. 2023;...]

  Sparse training methods

  [So many works, just look around you...]

Motivation for this work: End-2-end sparse training



- This work is part of a bigger "journey"
- *"Is it possible to train big models by training smaller versions of them?"*
- Past work answered this question affirmatively:

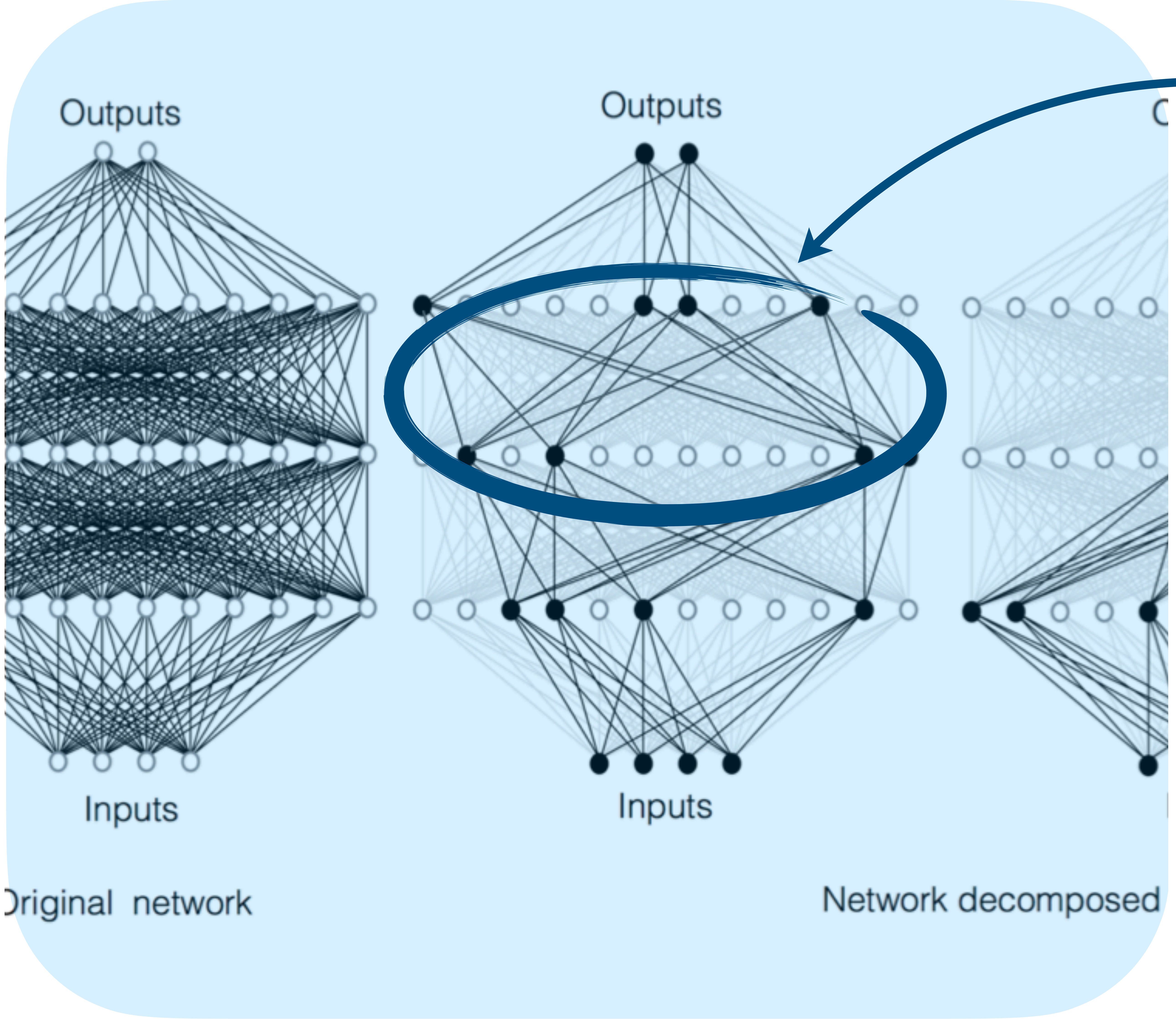
  Advertising own work: IST

  [Yuan et al. 2022, Liao et al 2021, Dun et al., 2022-23; Wan et al. 2022, Hu et al., 2023, Wang et al. 2023;...]

  Sparse training methods

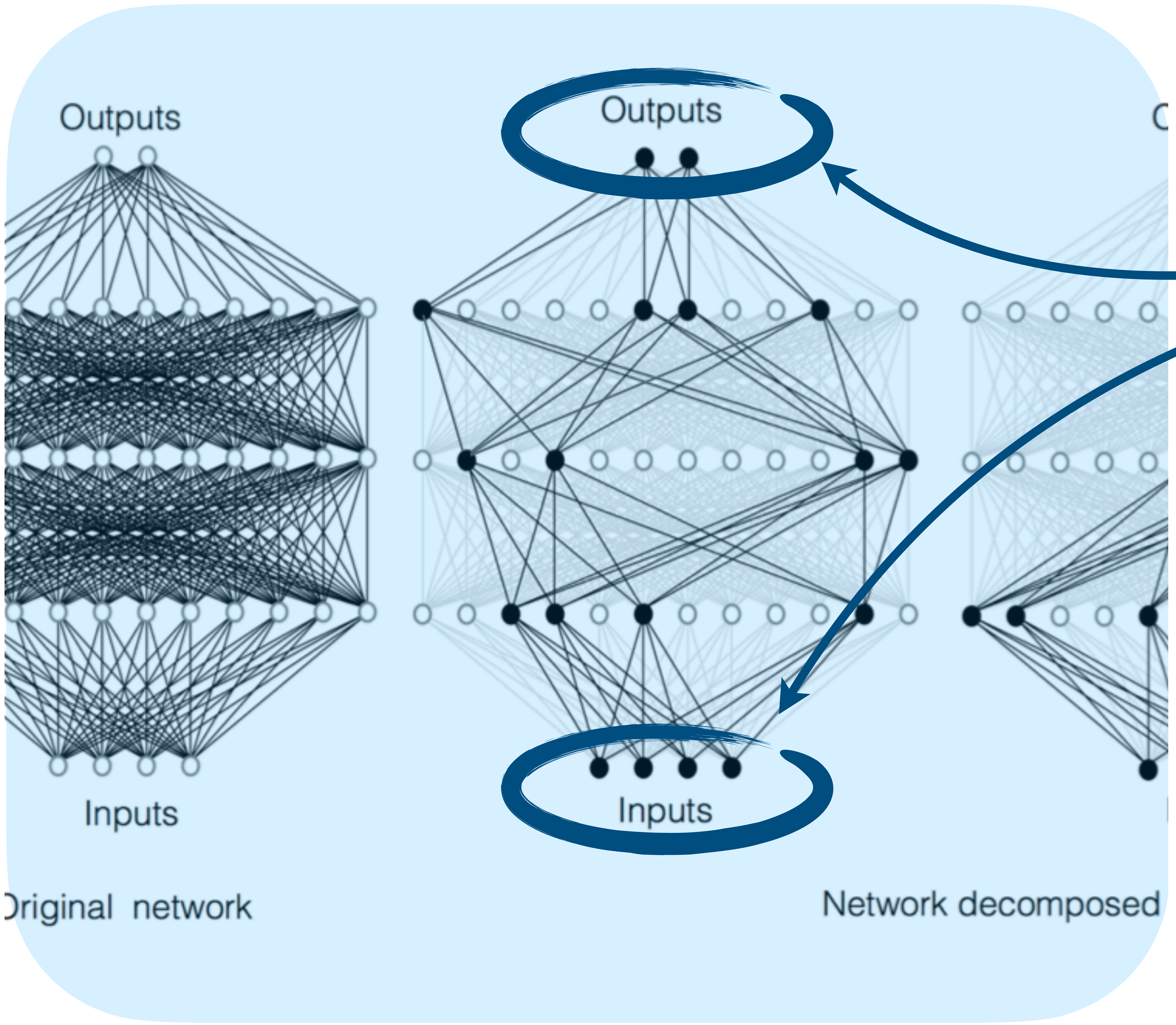
  [So many works, just look around you...]
- Motivation: efficiency; "do we need all these parameters?"; curiosity..

How does this work differentiate?



- Past work has focused on "inner" model sparsity (e.g., hidden layers)

How does this work differentiate?



- Past work has focused on “inner” model sparsity (e.g., hidden layers)
- There are cases where input/output layers create the bottleneck:

Number of classes in original ImageNet: 21K

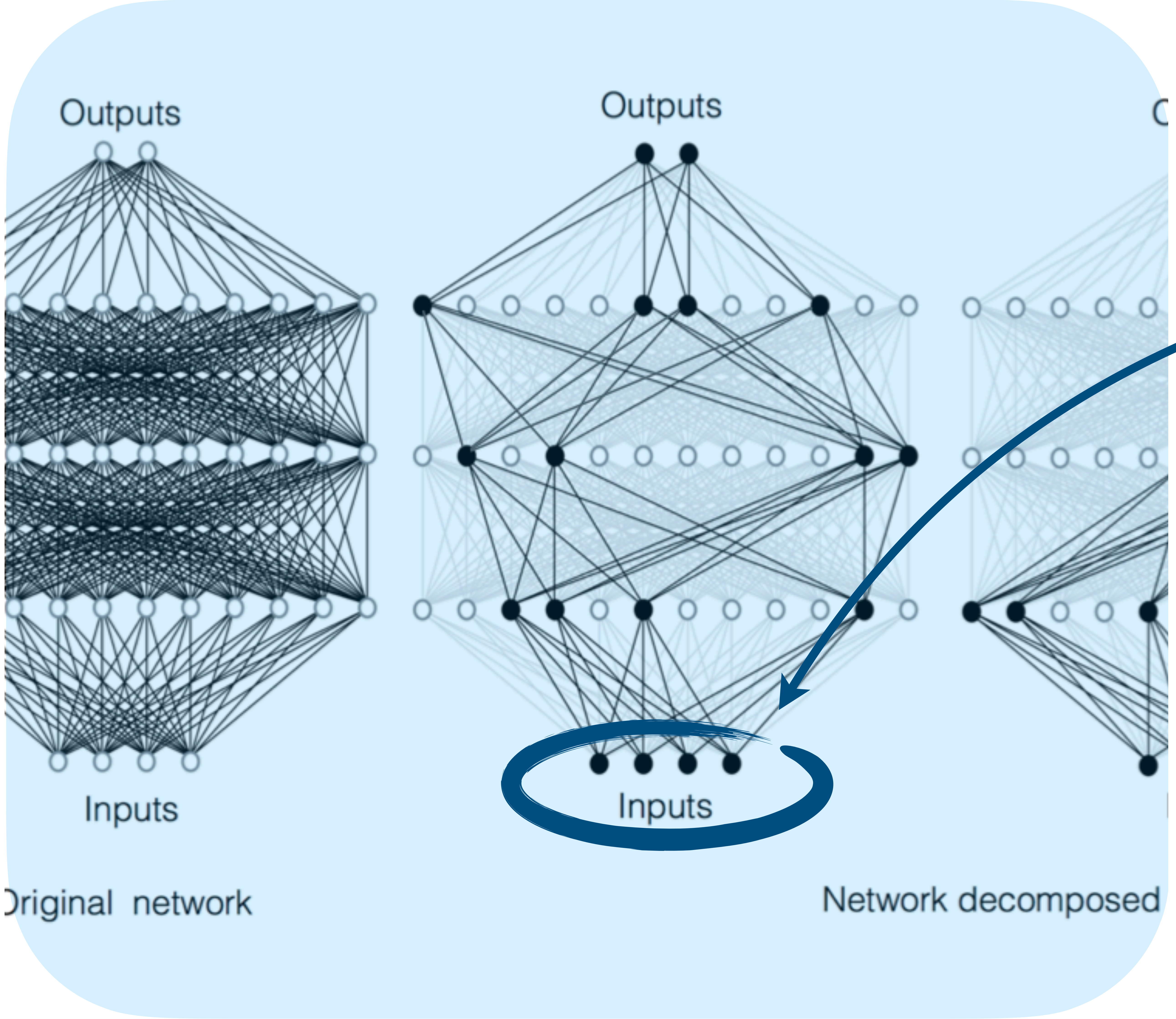
Number of products in recomm. datasets: Amazon 670K

Number of inputs: Amazon 670K has 135K input feature

32K token input for GPT-4?

.... (many more examples)

How does this work differentiate?



- Past work has focused on "inner" model sparsity (e.g., hidden layers)
- There are cases where input/output layers create the bottleneck:

  Number of classes in original ImageNet: 21K

  Number of products in recomm. datasets: Amazon 670K

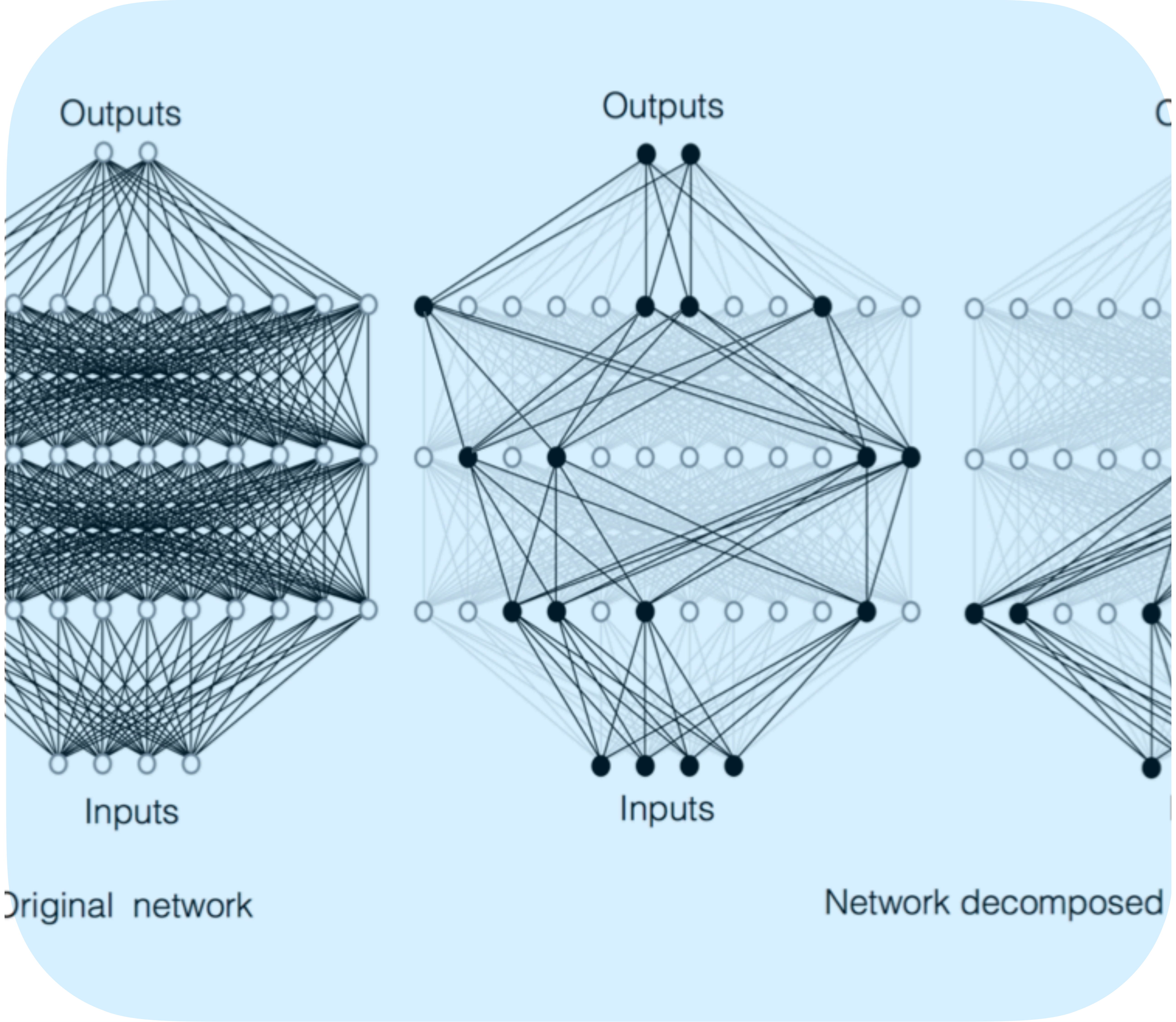
  Number of inputs: Amazon 670K has 135K input feature

  32K token input for GPT-4?

  .... (many more examples)

- In this work: we focus on input sparsity and combine this with other sparse-training methods (here IST - more later)

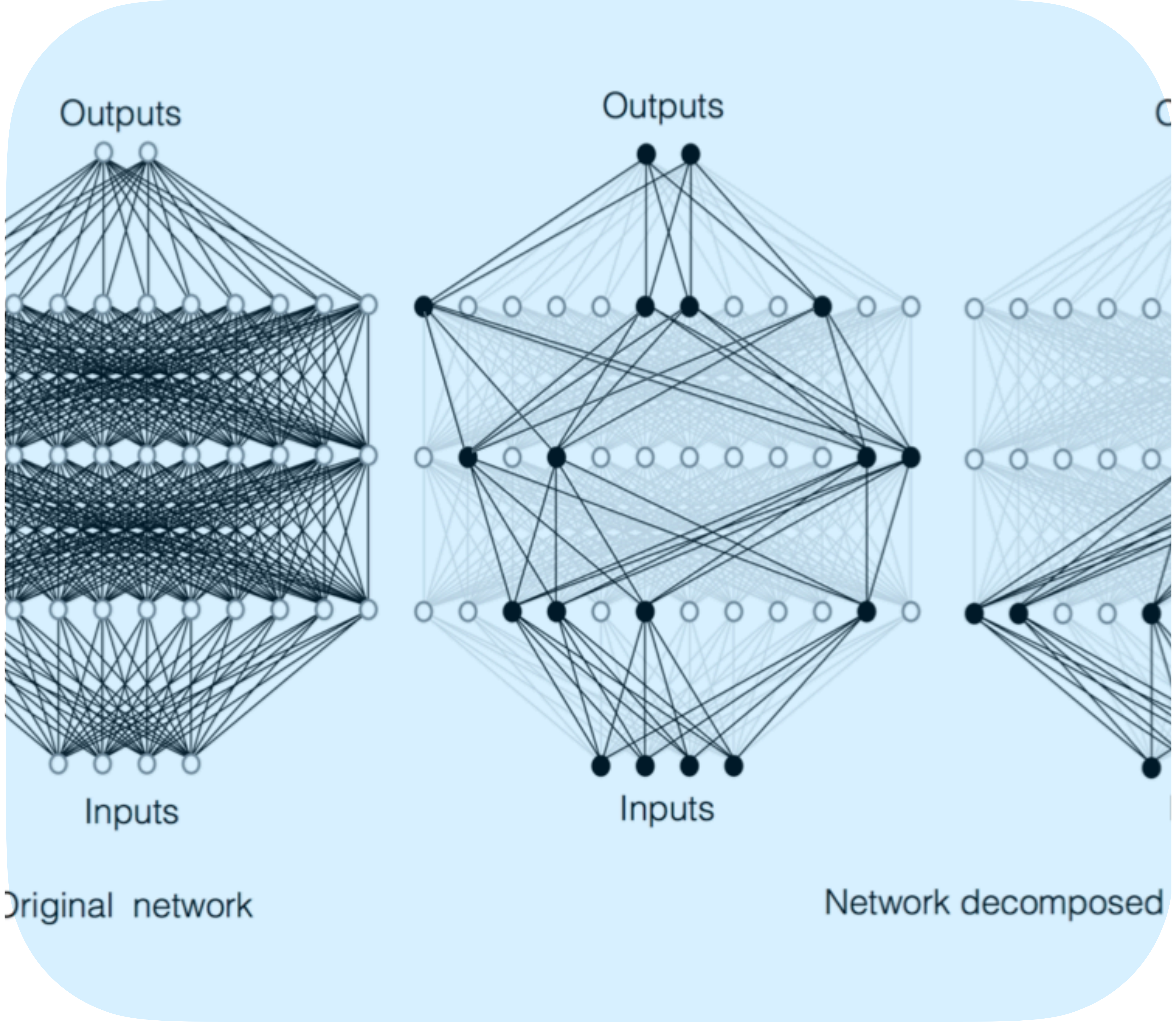
What is the scenario we consider?



- Computer vision (image classification)

  Other models (e.g., NLP): future directions

What is the scenario we consider?



- Computer vision (image classification)

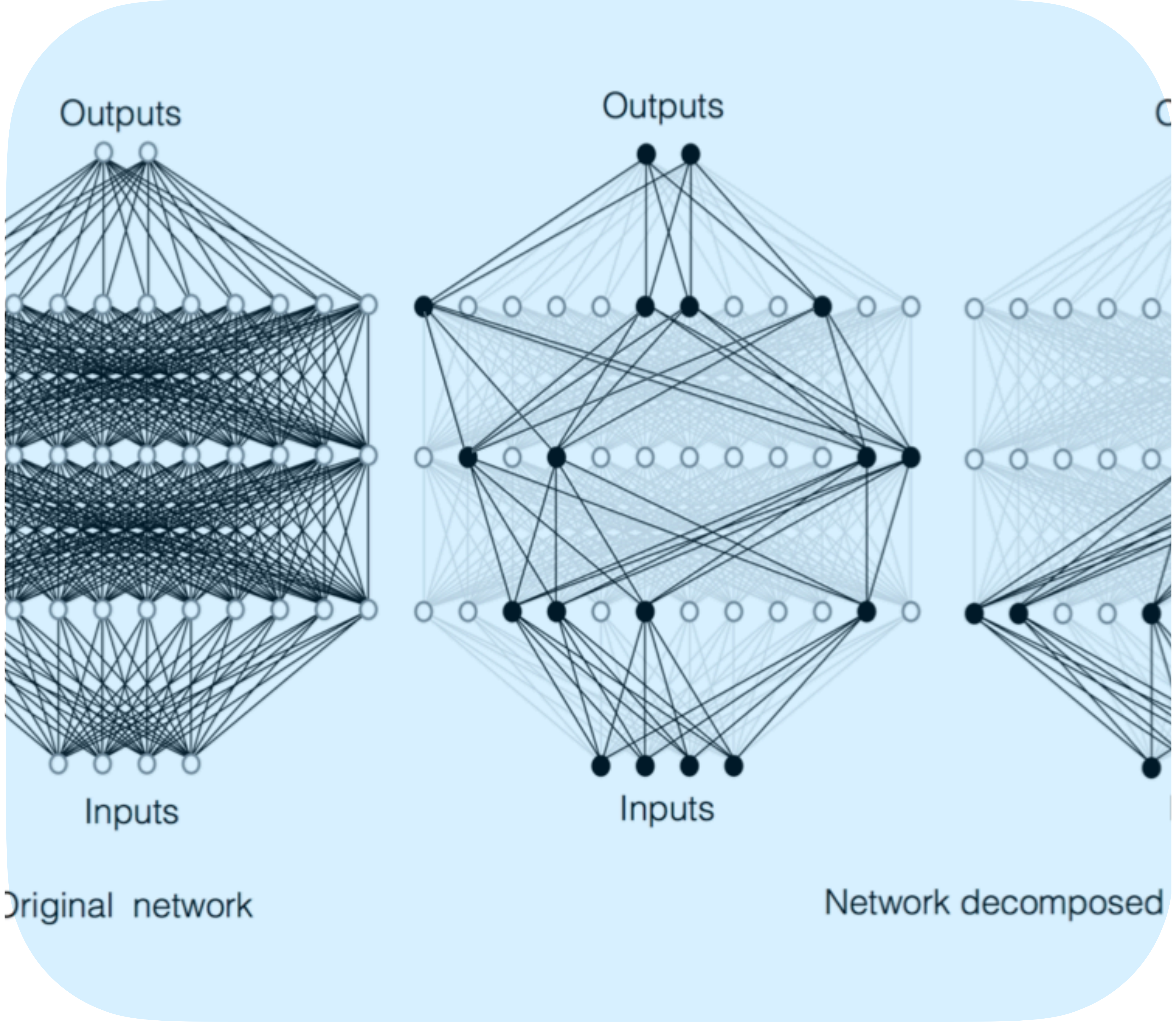
  Other models (e.g., NLP): future directions

- Reason for input sparsity:

  Either by nature

  Satellite remote sensing: cloud coverage limitations
  Patient movement in medical imaging

What is the scenario we consider?



- Computer vision (image classification)

  Other models (e.g., NLP): future directions

- Reason for input sparsity:

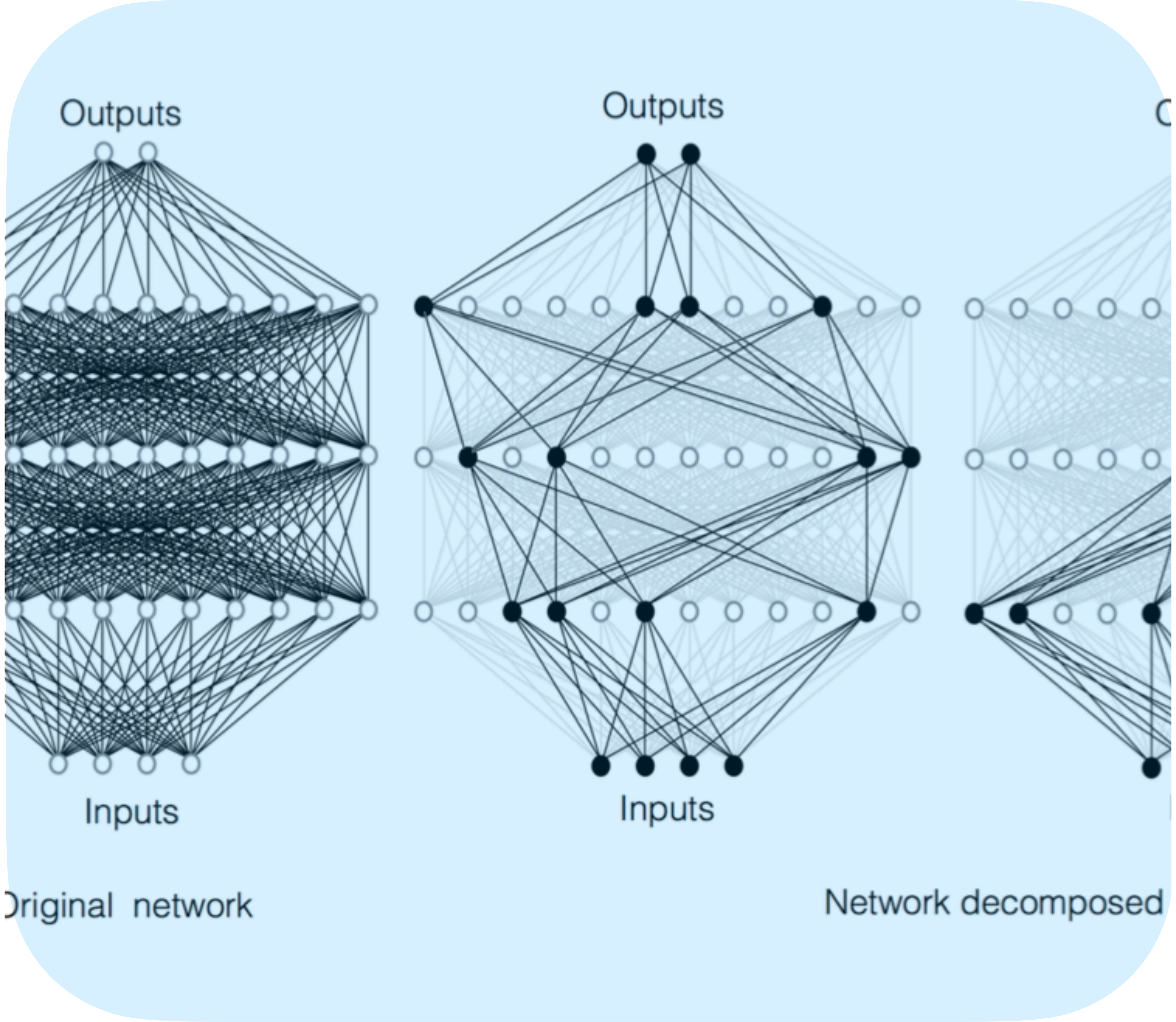
  Either by nature

  Satellite remote sensing: cloud coverage limitations
  Patient movement in medical imaging

  Or intentionally (for efficiency)

  "Sparse in enough for scaling transformers"
  "Masked auto encoders are scalable vision learners"
  ...

What is the scenario we consider?



- Computer vision (image classification)

    Other models (e.g., NLP): future directions

- Reason for input sparsity:

    Either by nature

    Satellite remote sensing: cloud coverage limitations
    Patient movement in medical imaging

    Or intentionally (for efficiency)

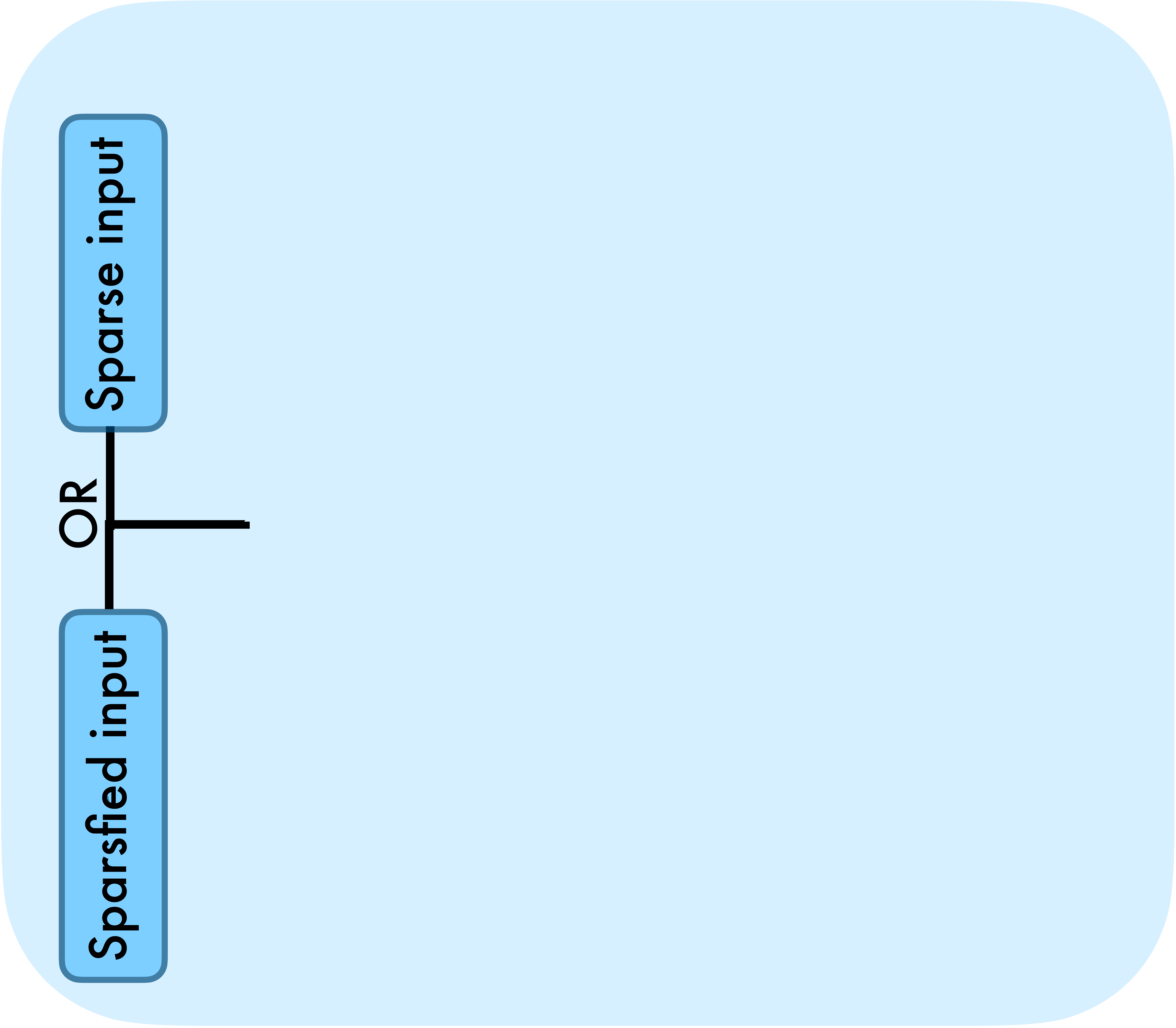
    "Sparse in enough for scaling transformers"
    "Masked auto encoders are scalable vision learners"
    …

- Consider also a distributed scenario

    Either regular distributed or FL

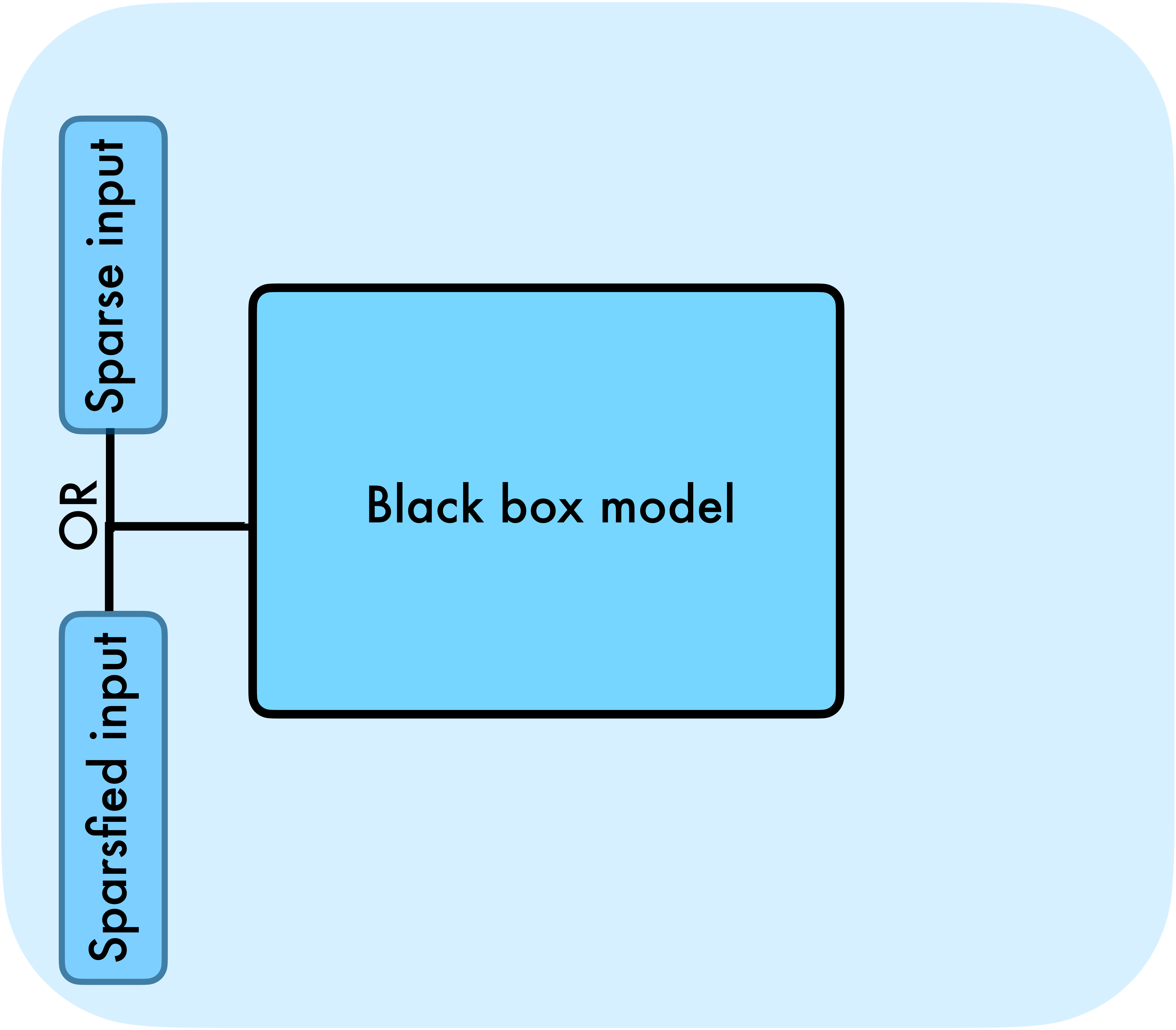
    Not much power to do intensive work
    Not much memory to keep large models

How does the model look like so far?



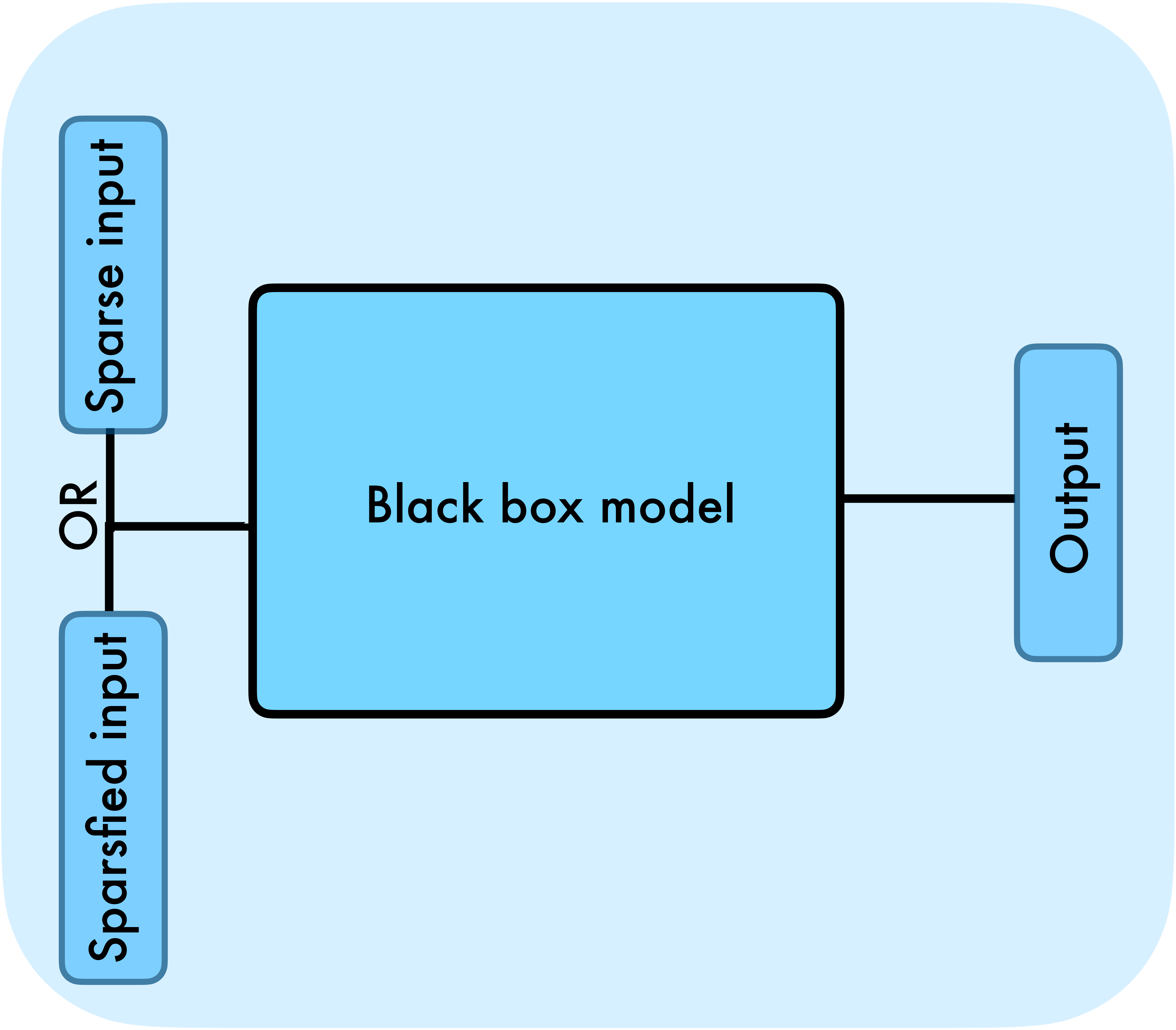
- Sparse input

How does the model look like so far?



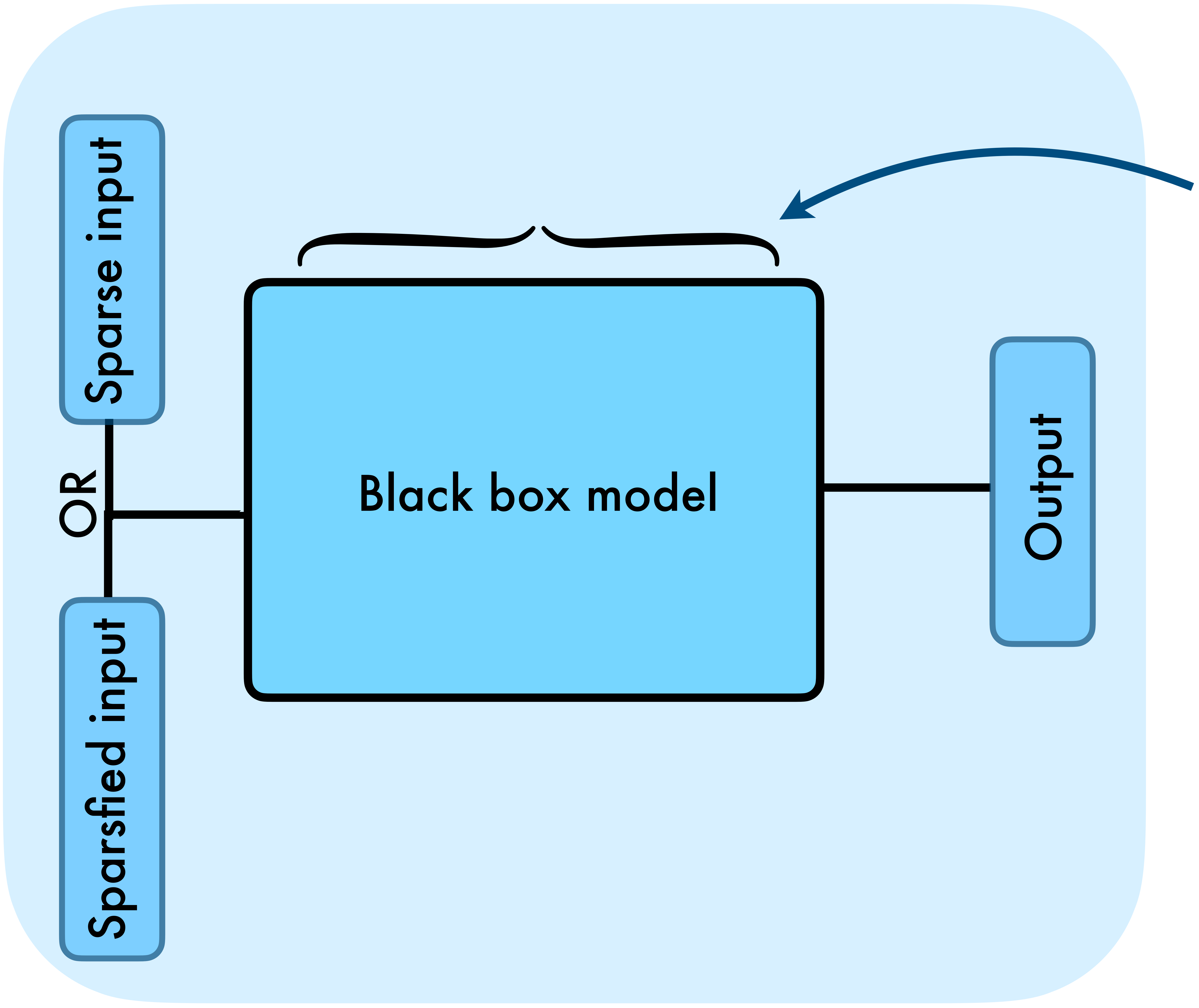
- Sparse input

How does the model look like so far?



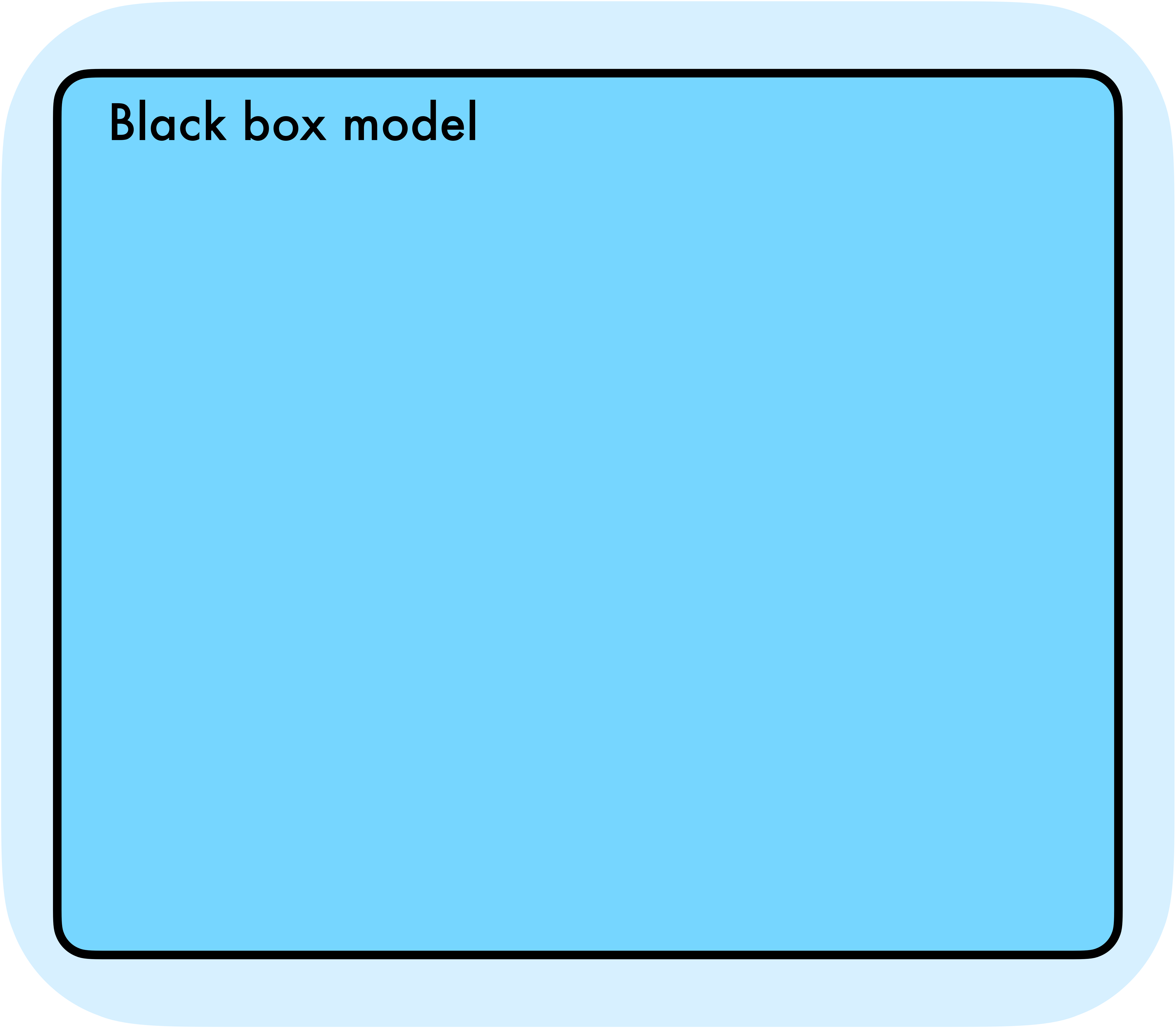
- Sparse input

How does the model look like so far?



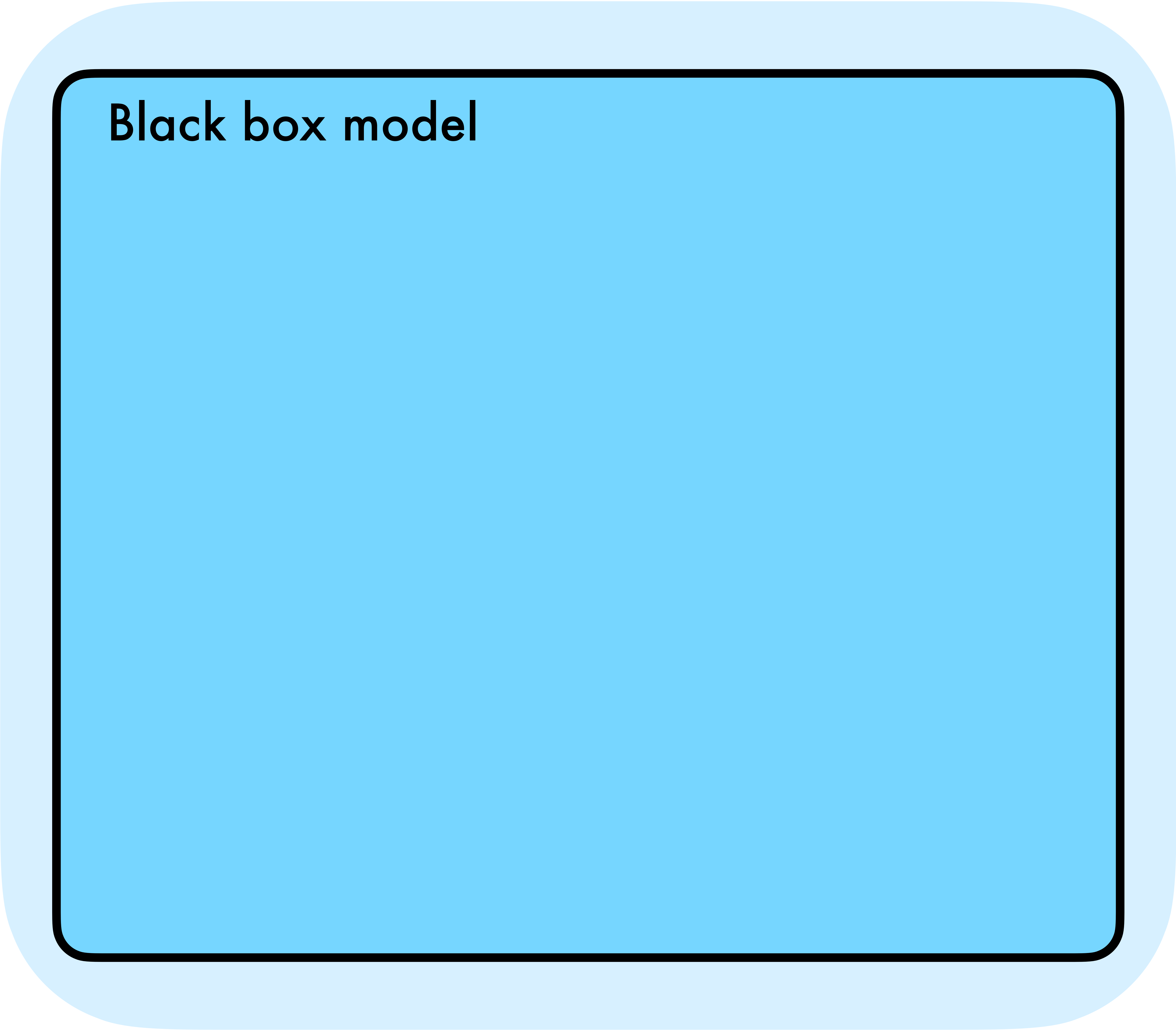
- Sparse input
- What about the intermediate layers?

How does the model look like so far?



- "Zooming in" the black box model...

How does the model look like so far?



- “Zooming in” the black box model...
- Disclaimer: our goal is to have end-2-end sparse training

How does the model look like so far?

Black box model

(Could have been a fully sparse NN modeling + training procedure but...)

- "Zooming in" the black box model...
- Disclaimer: our goal is to have end-2-end sparse training

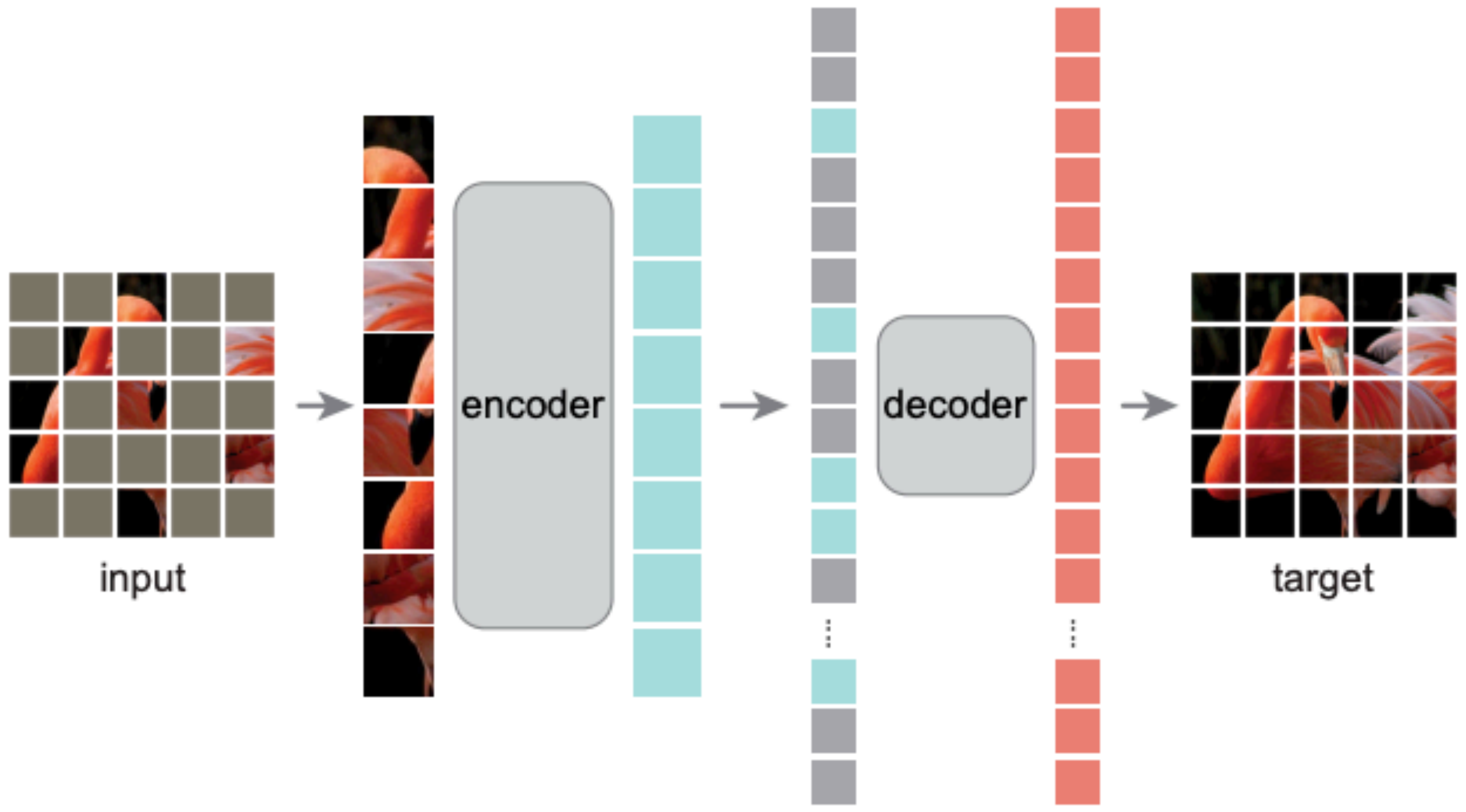
How does the model look like so far?

Black box model

(Could have been a fully sparse NN modeling + training procedure but...)

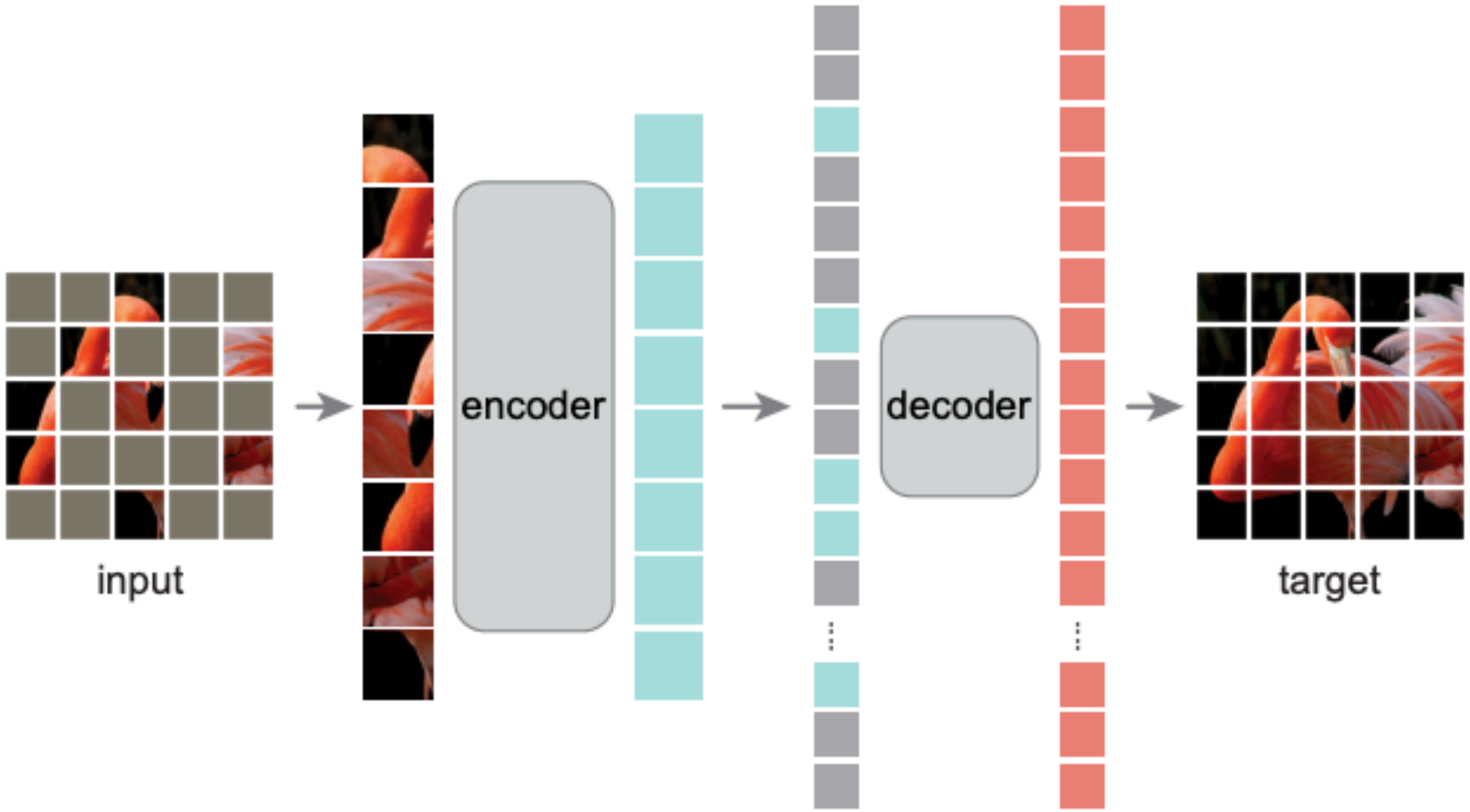
- "Zooming in" the black box model...
- Disclaimer: our goal is to have end-2-end sparse training
- But we cannot neglect pretrained models

Masked autoencoders (MAEs)



MAEs [He et al., 2021]

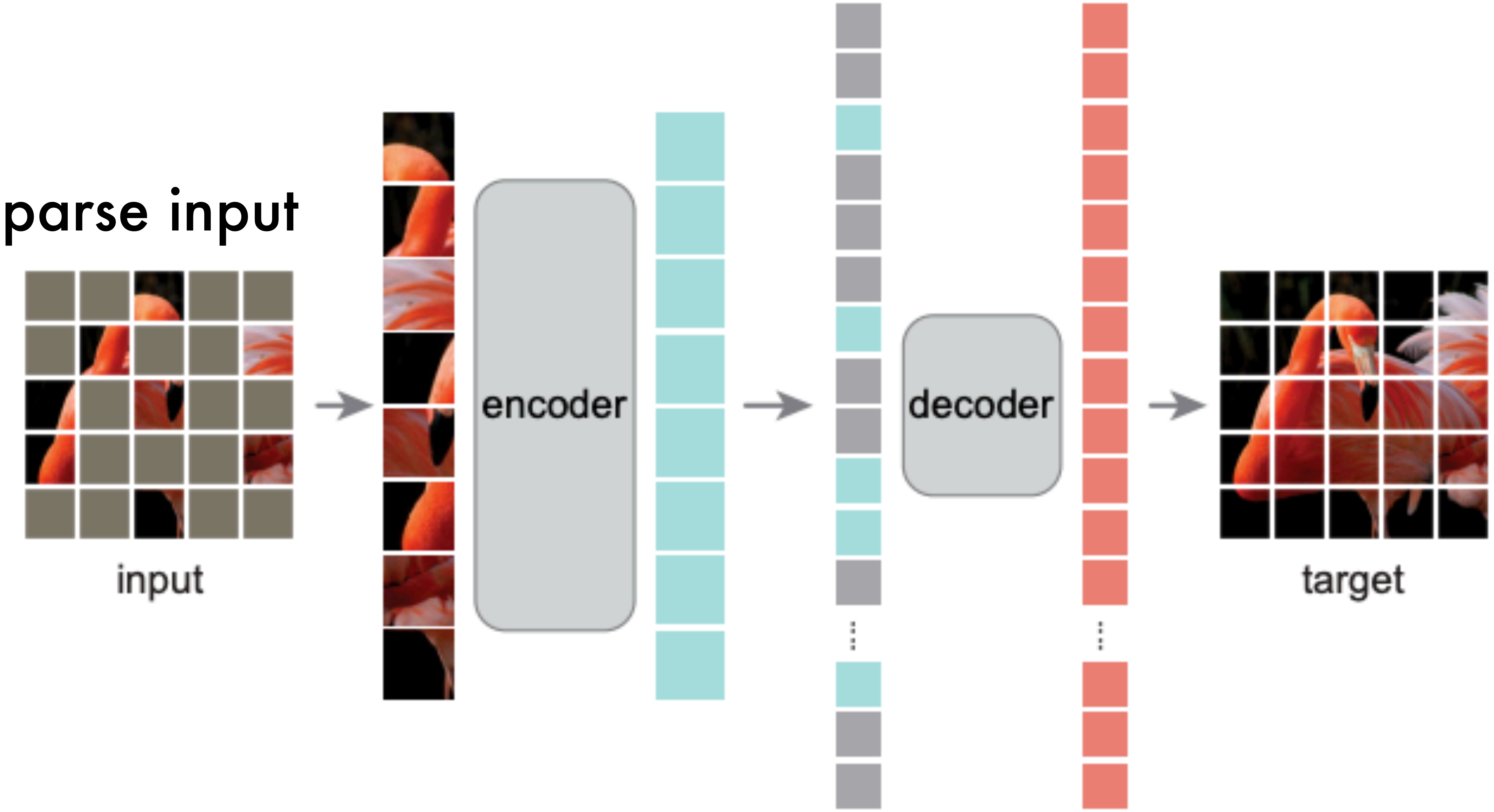
- The idea is that of regression (reconstruction) from sparse input (images)



MAEs [He et al., 2021]

- The idea is that of regression (reconstruction) from sparse input (images)

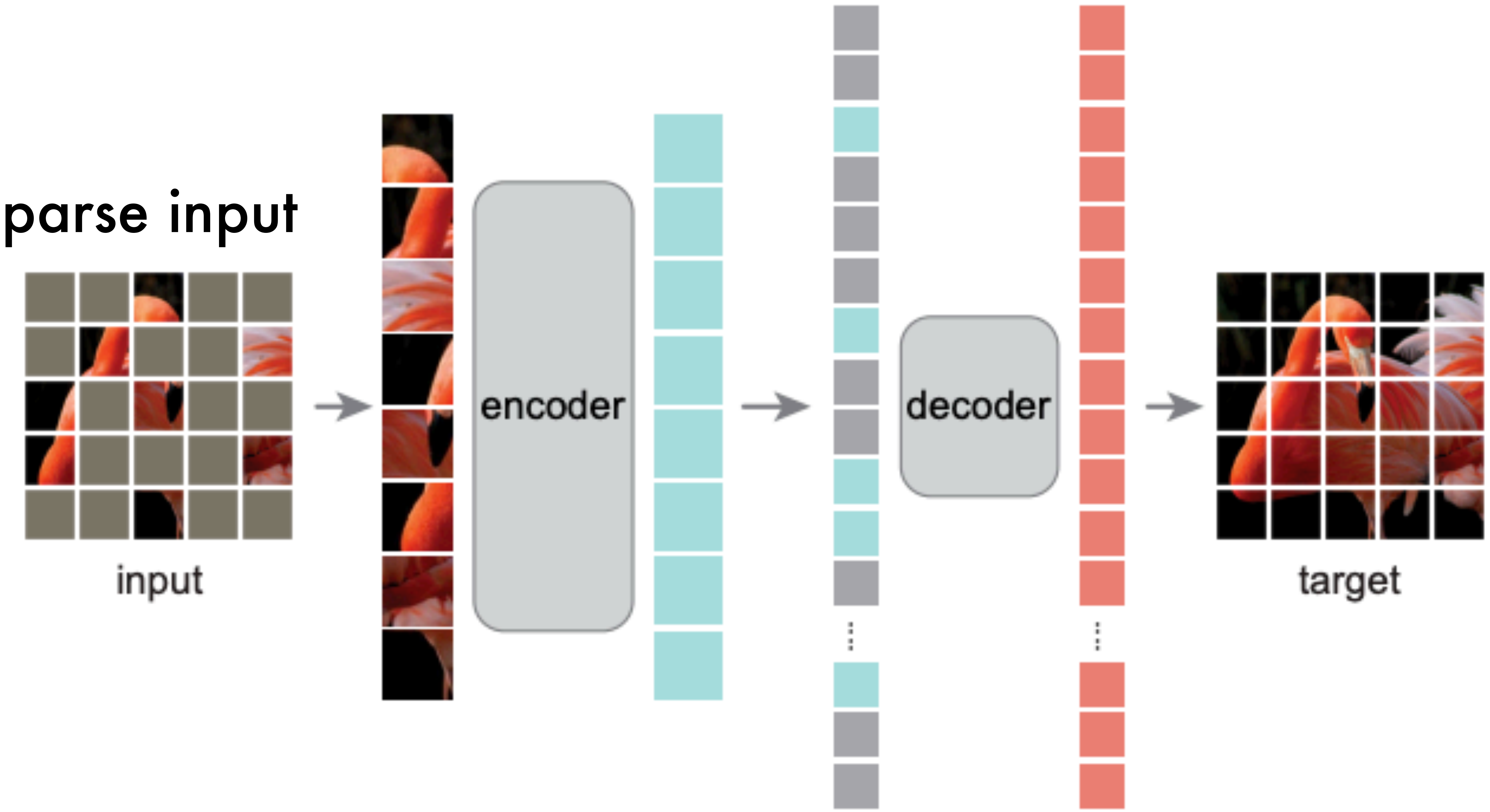
Sparse input



MAEs [He et al., 2021]

- The idea is that of regression (reconstruction) from sparse input (images)

Sparse input



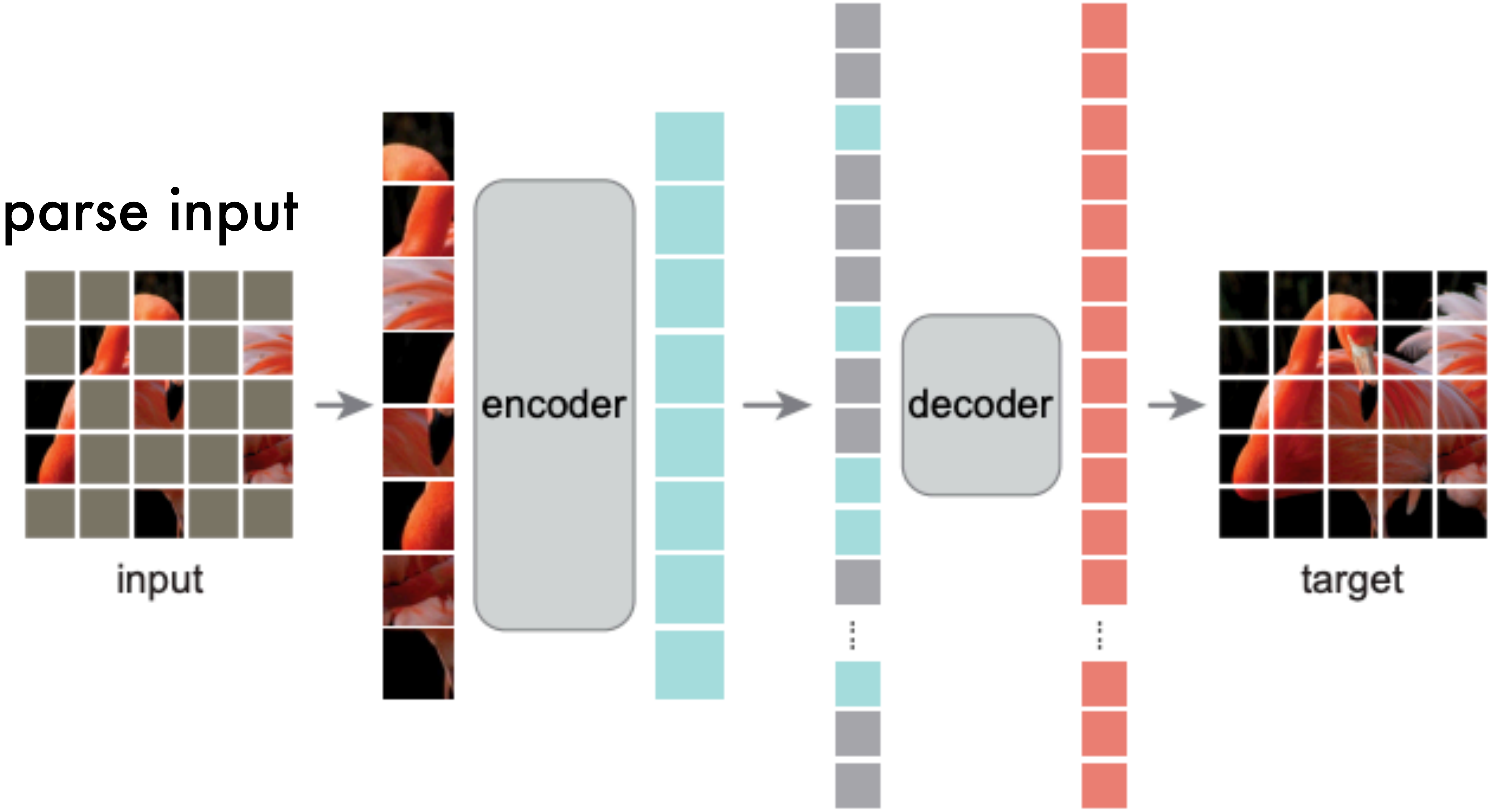
Feature extraction

MAEs [He et al., 2021]

- The idea is that of regression (reconstruction) from sparse input (images)

Decoding (image reconstruction)

Sparse input



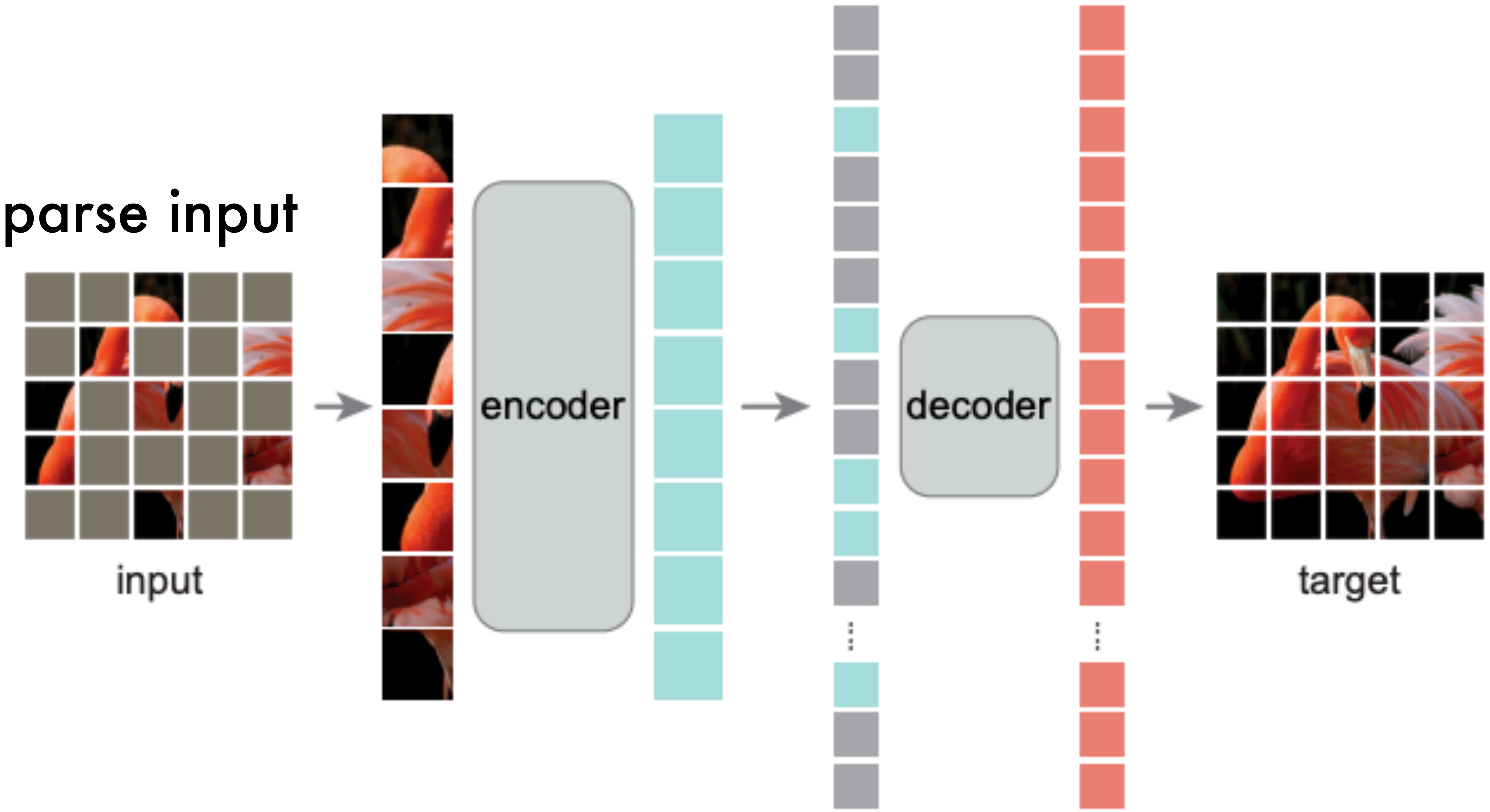
Feature extraction

MAEs [He et al., 2021]

- The idea is that of regression (reconstruction) from sparse input (images)
- The whole architecture is Transformer-based (ViT)
- Pretrained on ImageNet data

Decoding (image reconstruction)

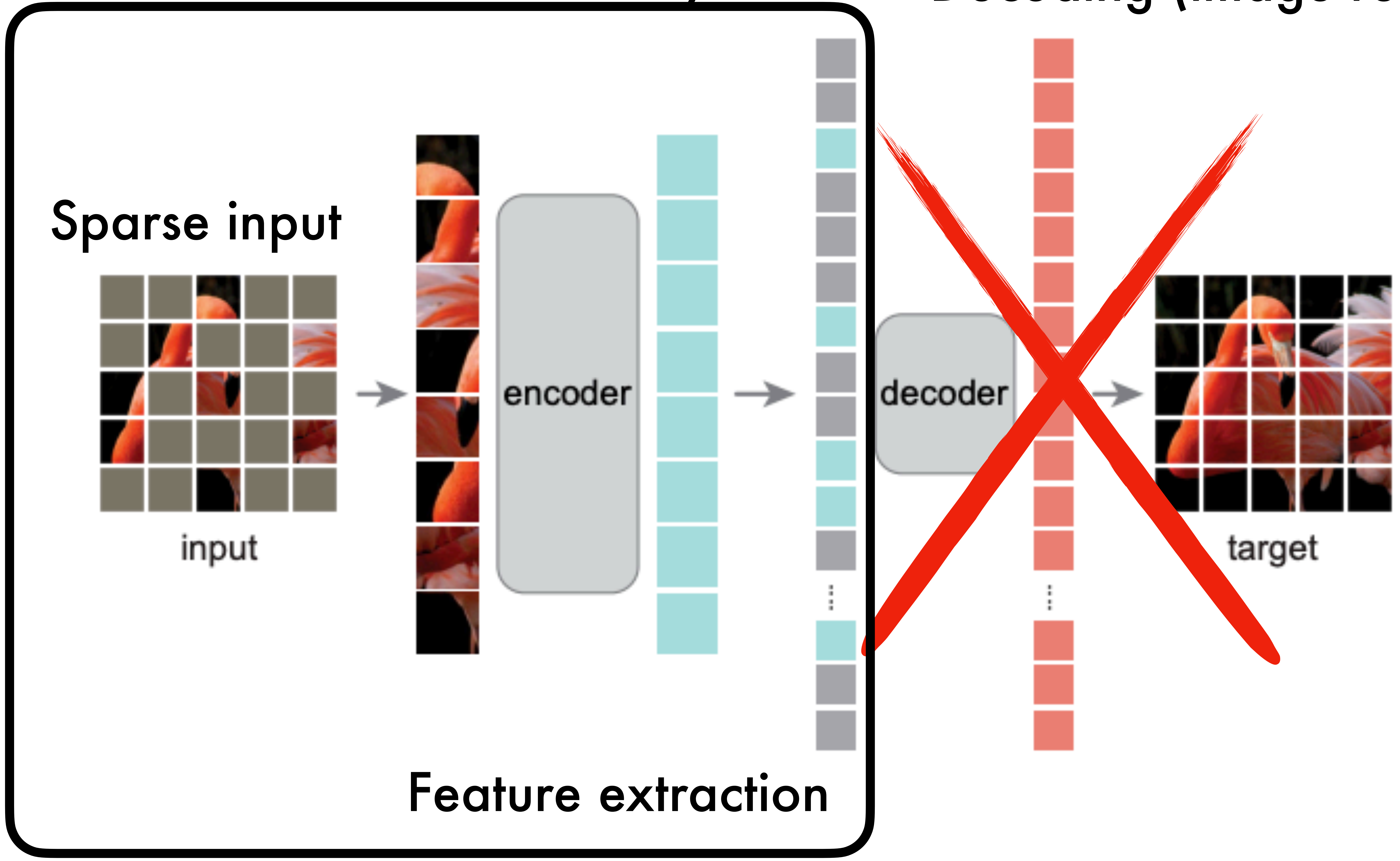
Sparse input



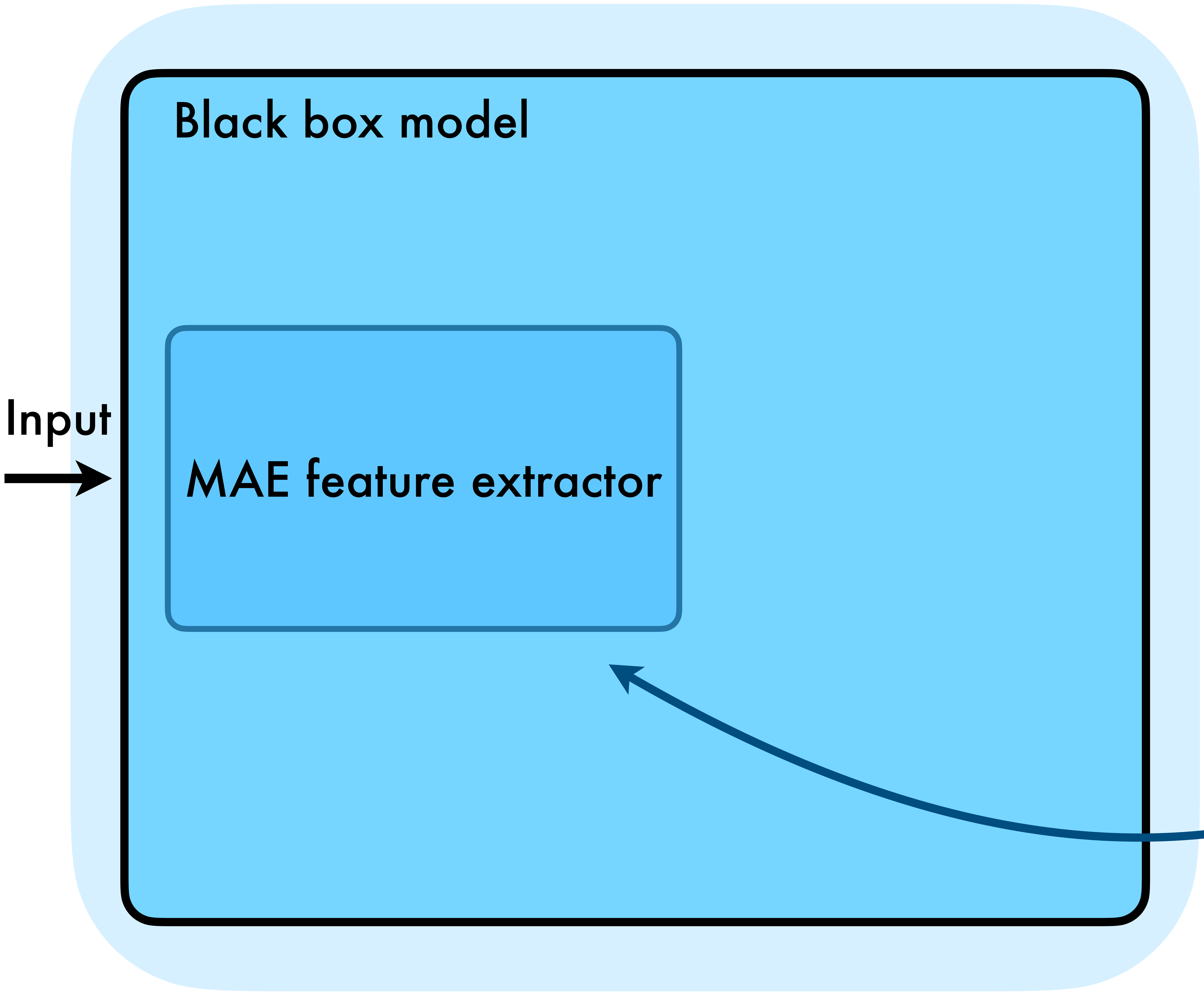
Feature extraction

MAEs [He et al., 2021]

- The idea is that of regression (reconstruction) from sparse input (images)
- The whole architecture is Transformer-based (ViT)
- Pretrained on ImageNet data
- In our case: we use the feature extractor only!

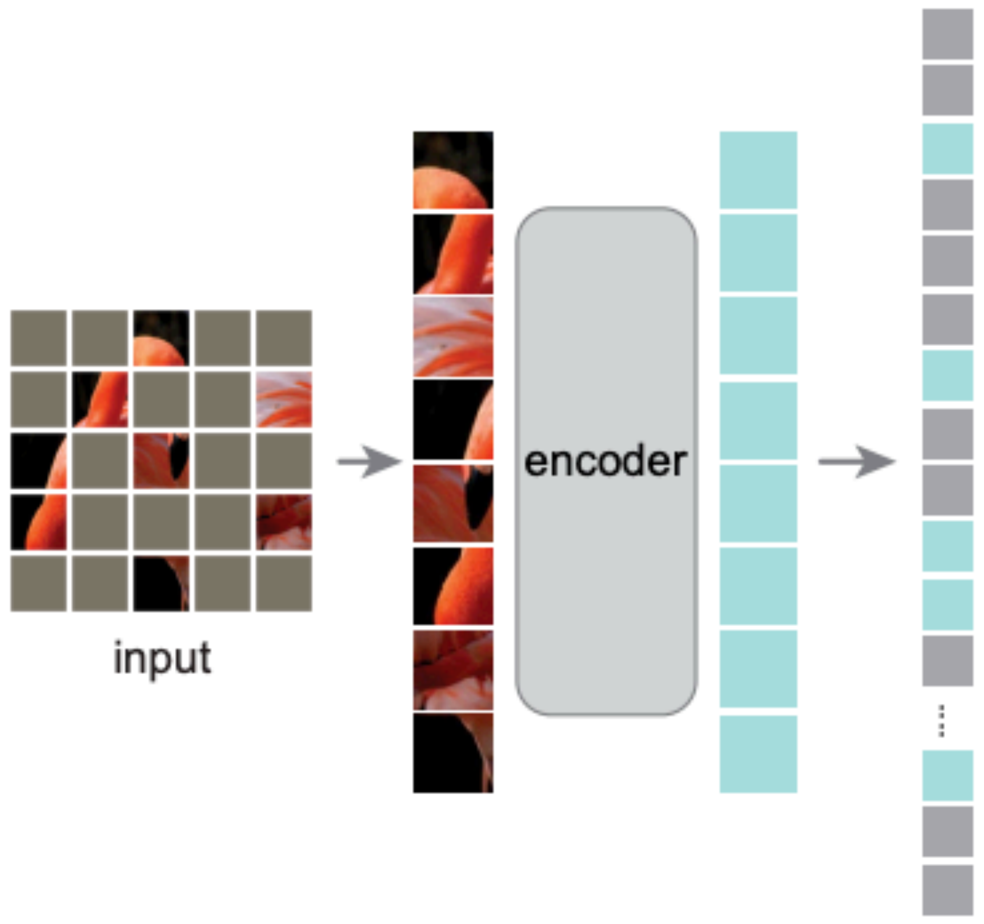


How does the model look like so far?

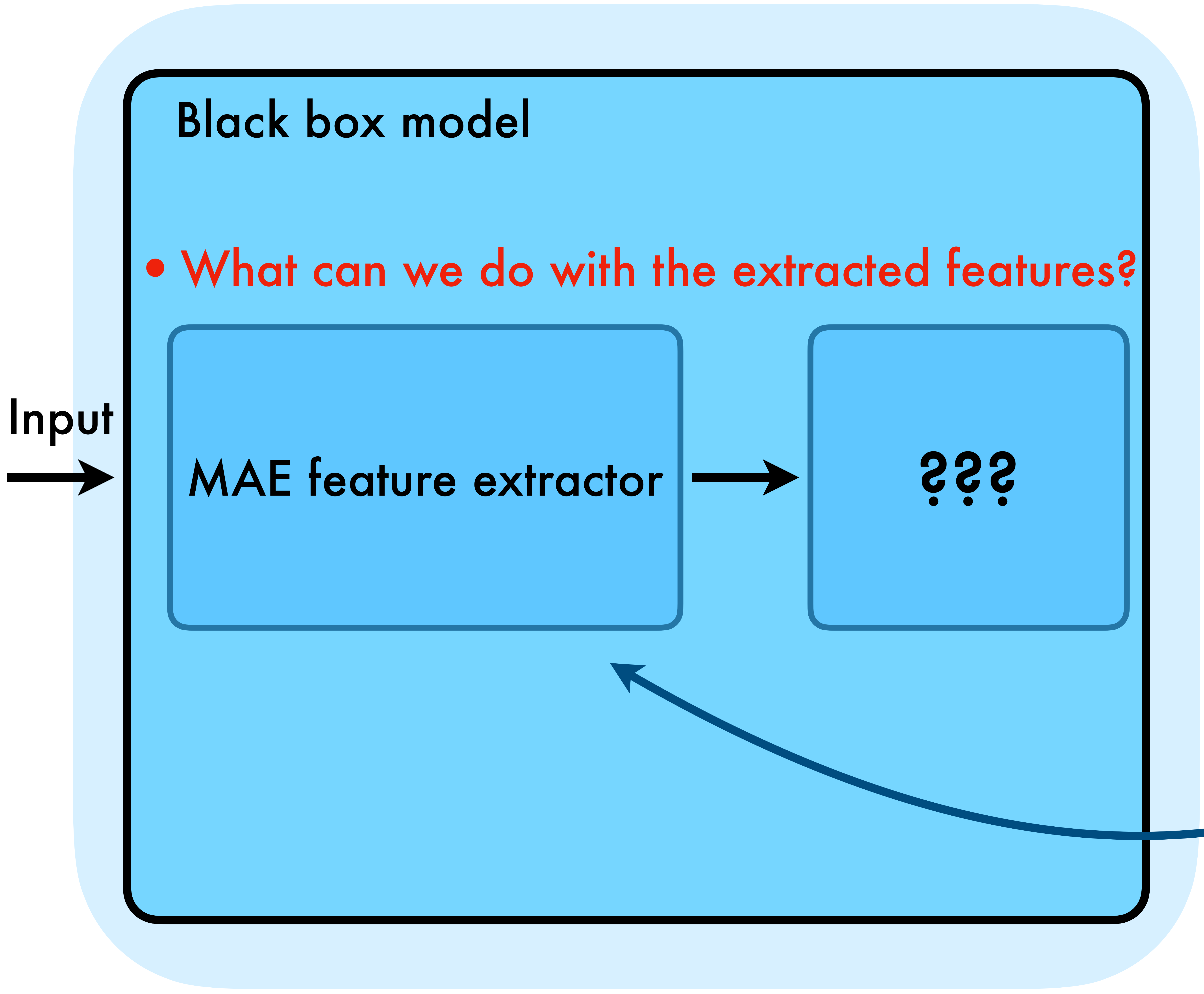


- "Zooming in" the black box model...
- Disclaimer: our goal is to have end-2-end sparse training
- But we cannot neglect pretrained models

Masked autoencoders (MAEs)

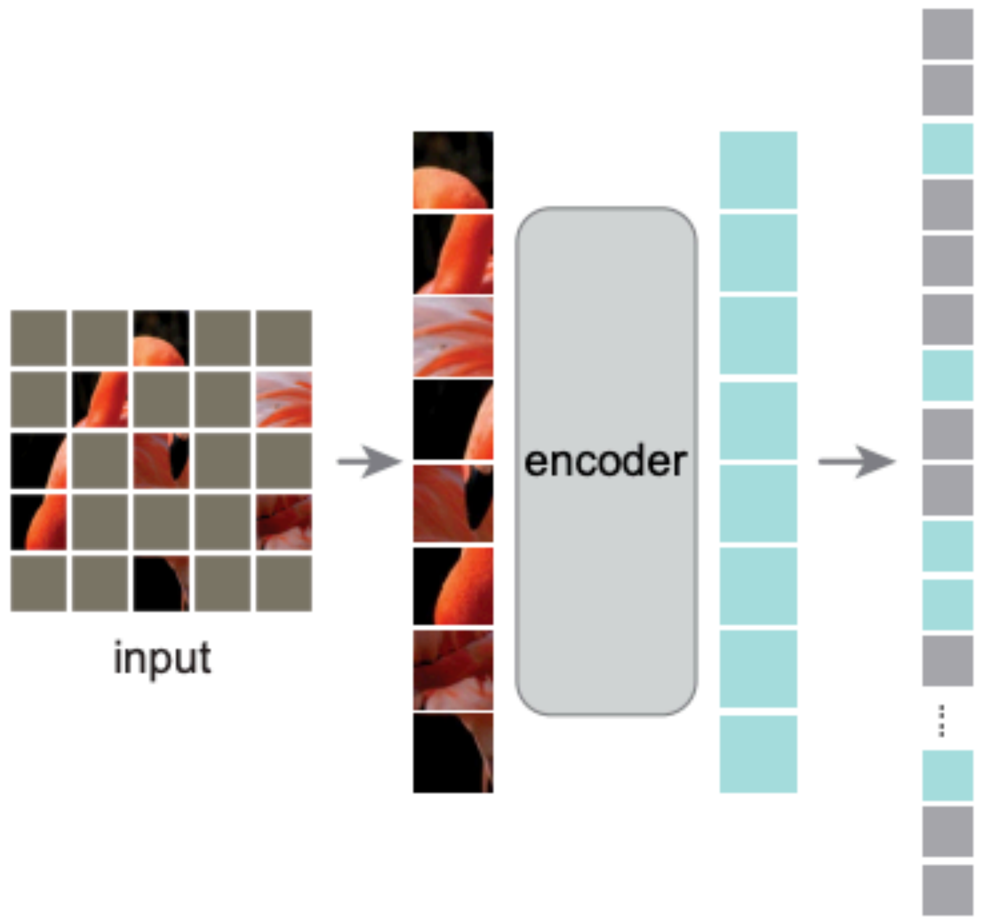


How does the model look like so far?

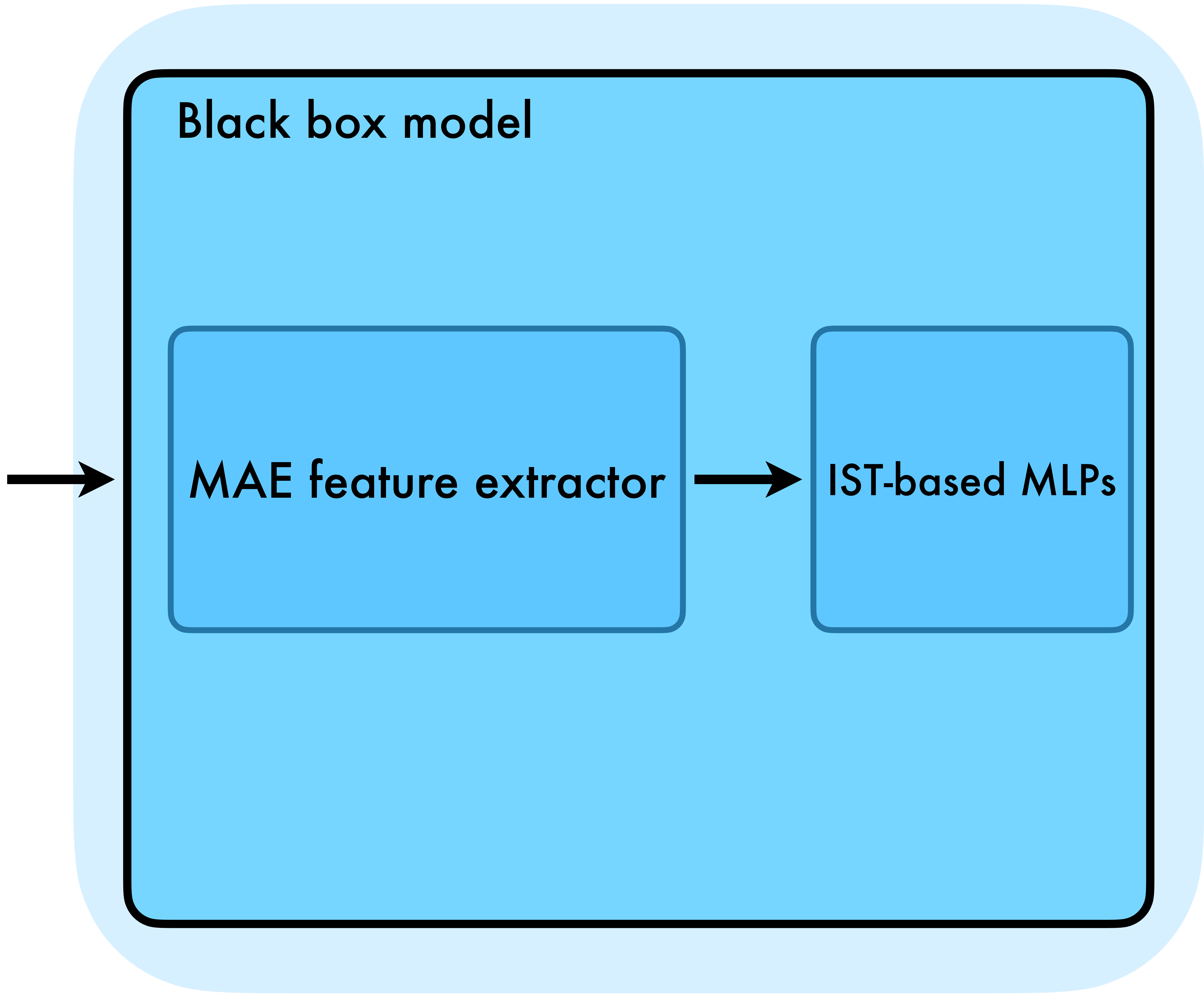


- "Zooming in" the black box model…
- Disclaimer: our goal is to have end-2-end sparse training
- But we cannot neglect pretrained models

Masked autoencoders (MAEs)

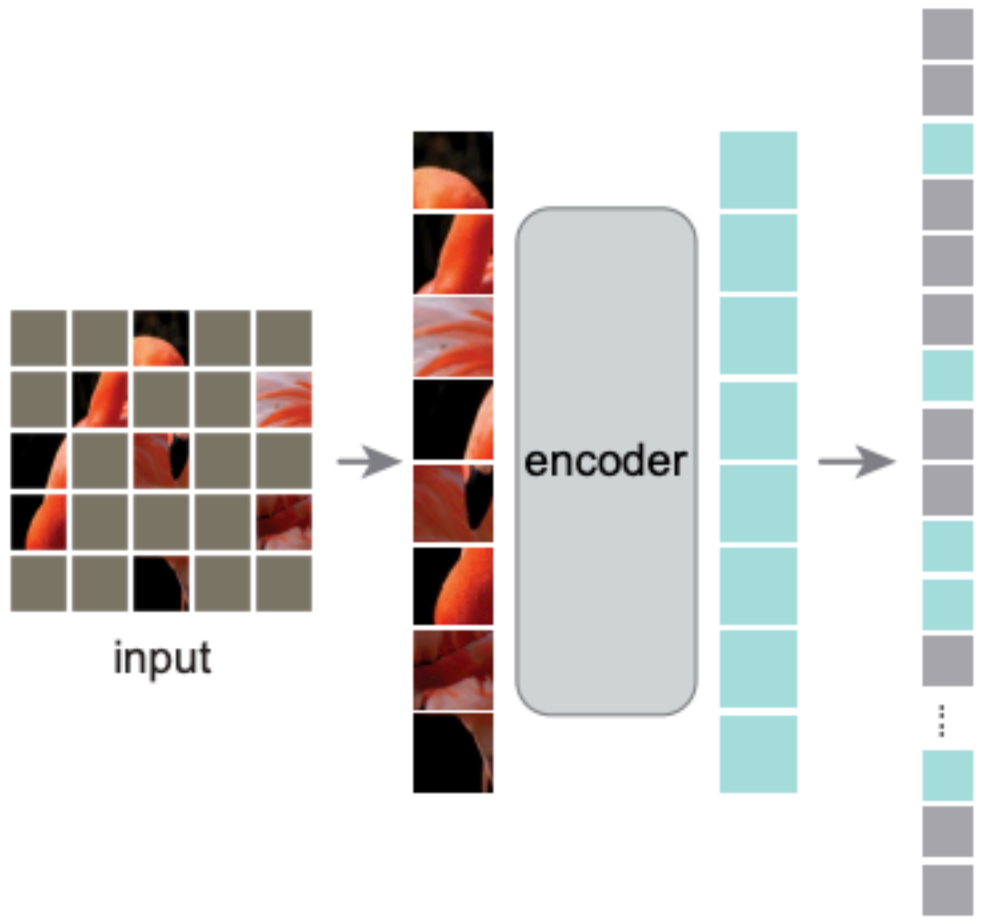


How does the model look like so far?

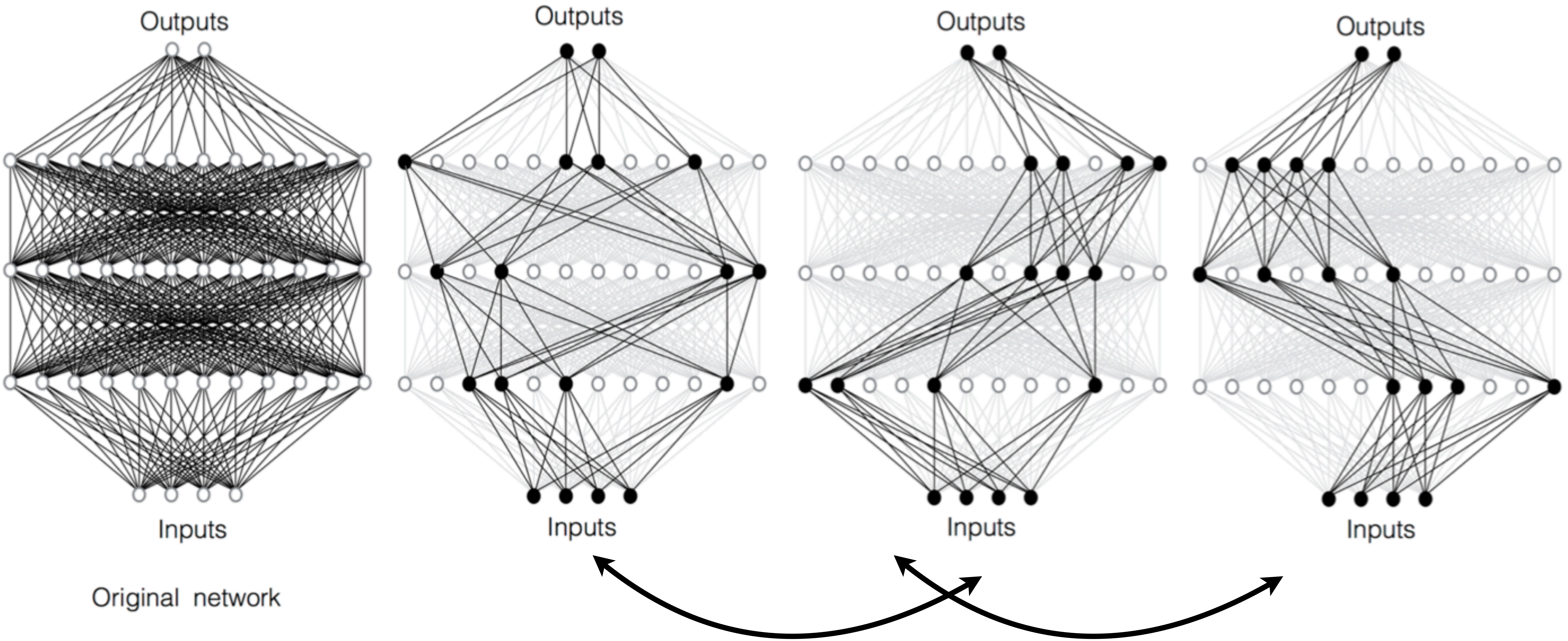


- "Zooming in" the black box model...
- Disclaimer: our goal is to have end-2-end sparse training
- But we cannot neglect pretrained models

Masked autoencoders (MAEs)



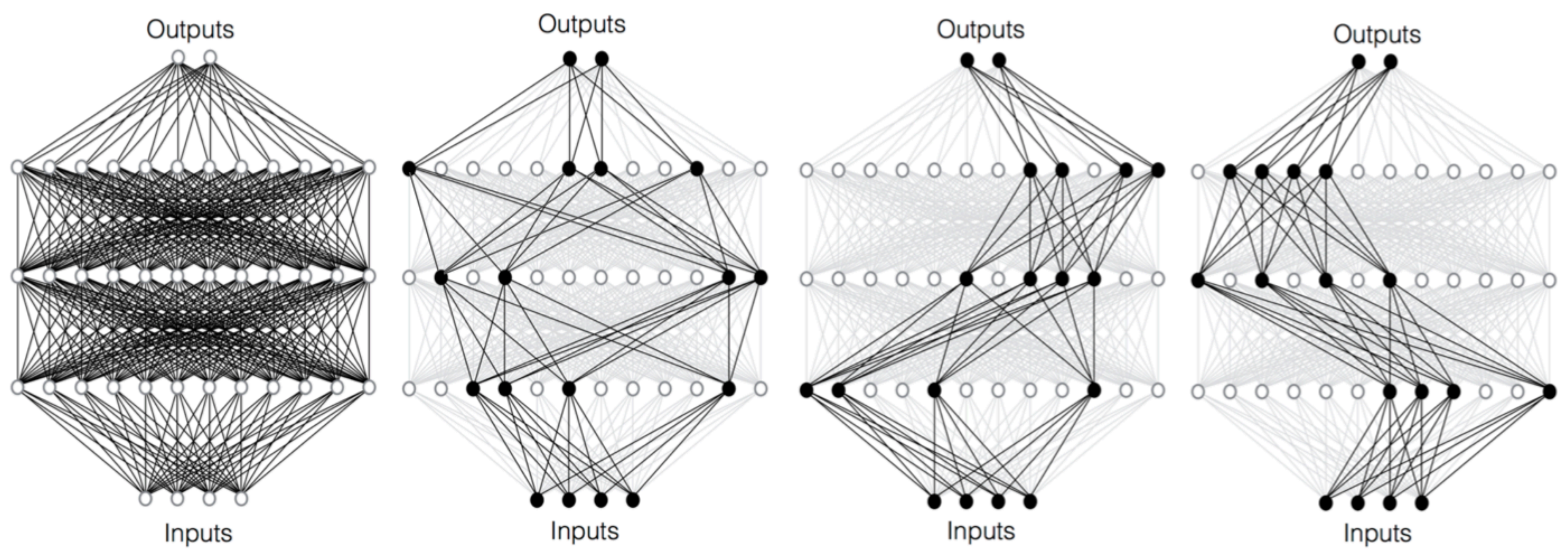
IST: Independent Subnet Training



Union of neurons make original network
(Note: union of parameters do not make original network necessarily)

IST: Independent Subnet Training

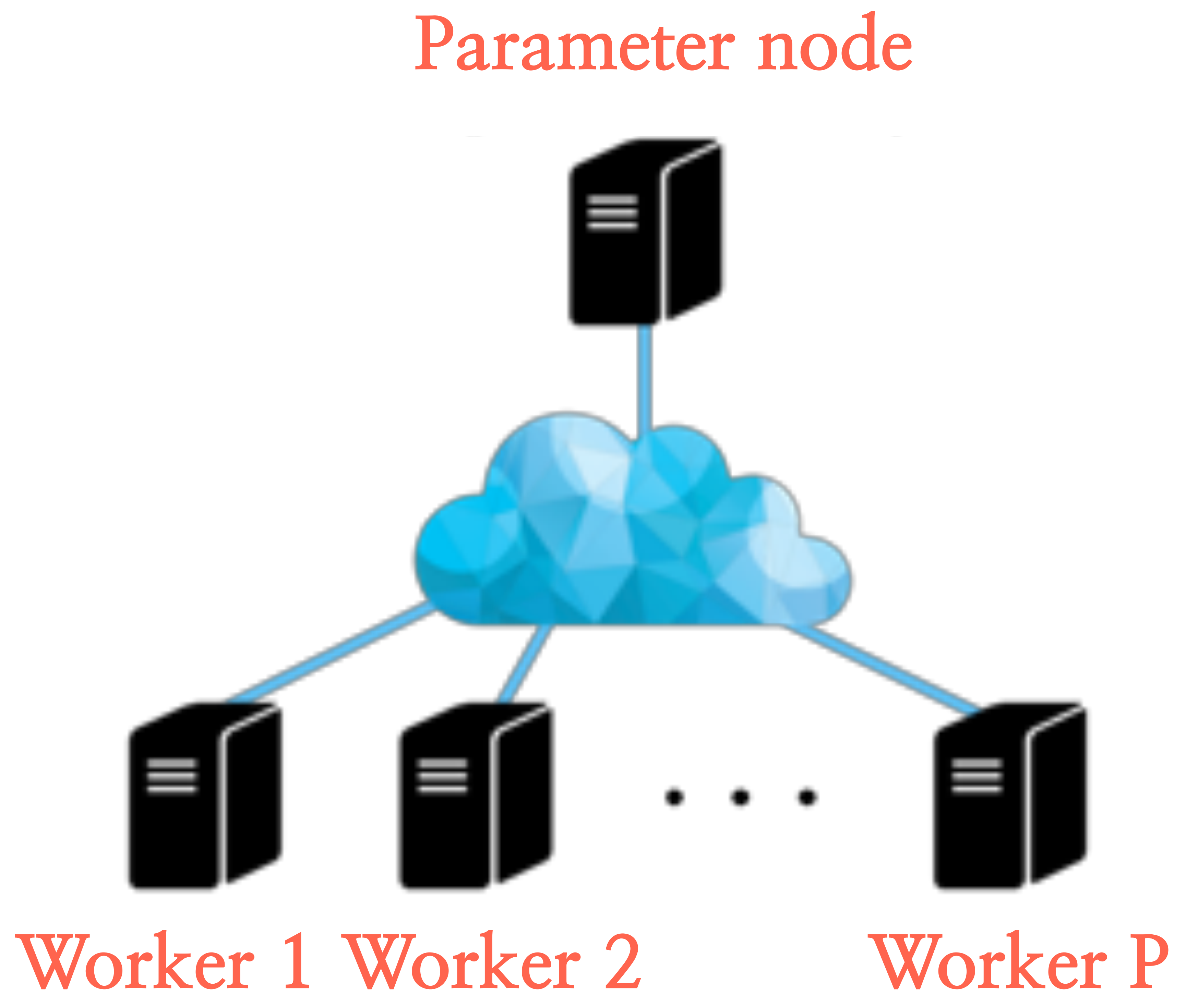
How to decompose a NN:



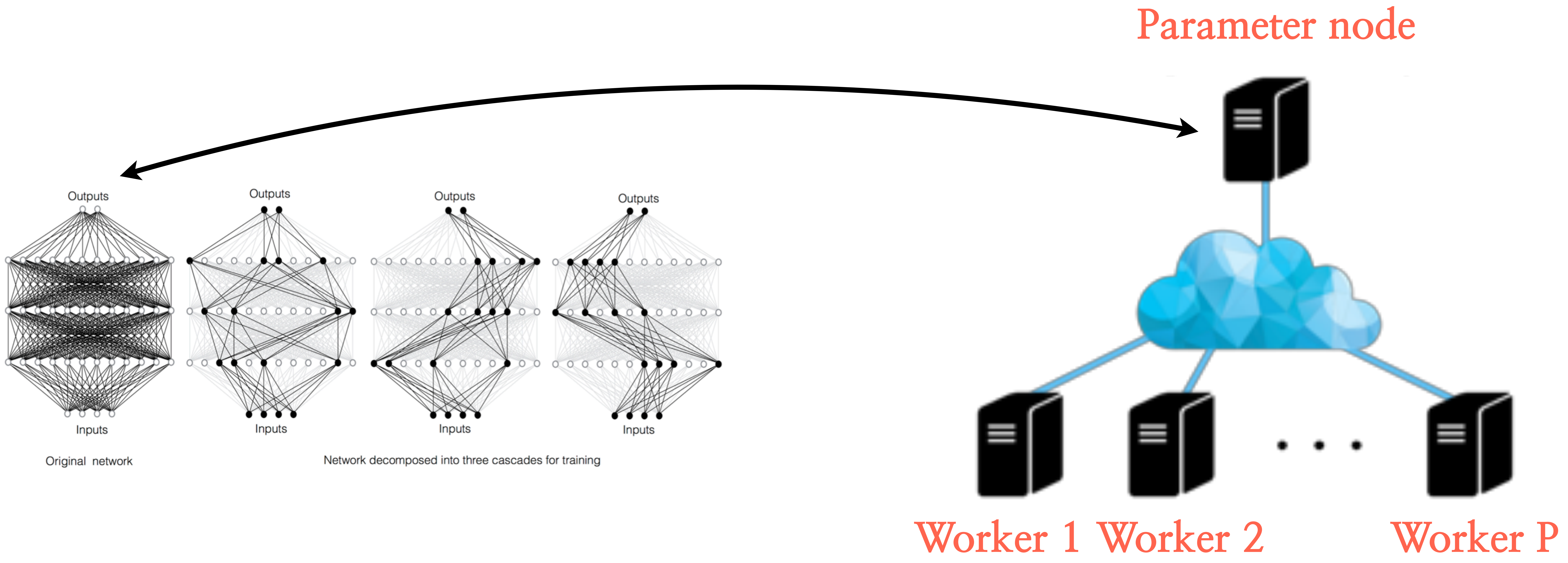
Original network

Network decomposed into three cascades for training

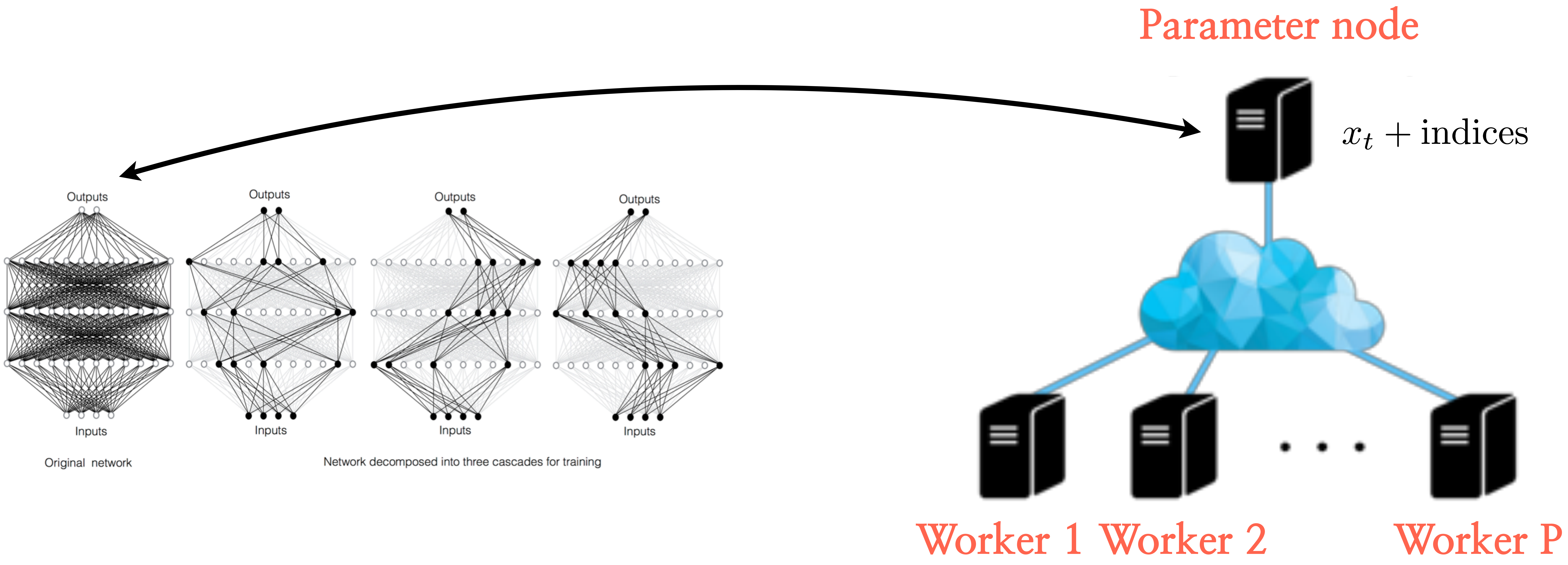
How to train NN in a distributed fashion:



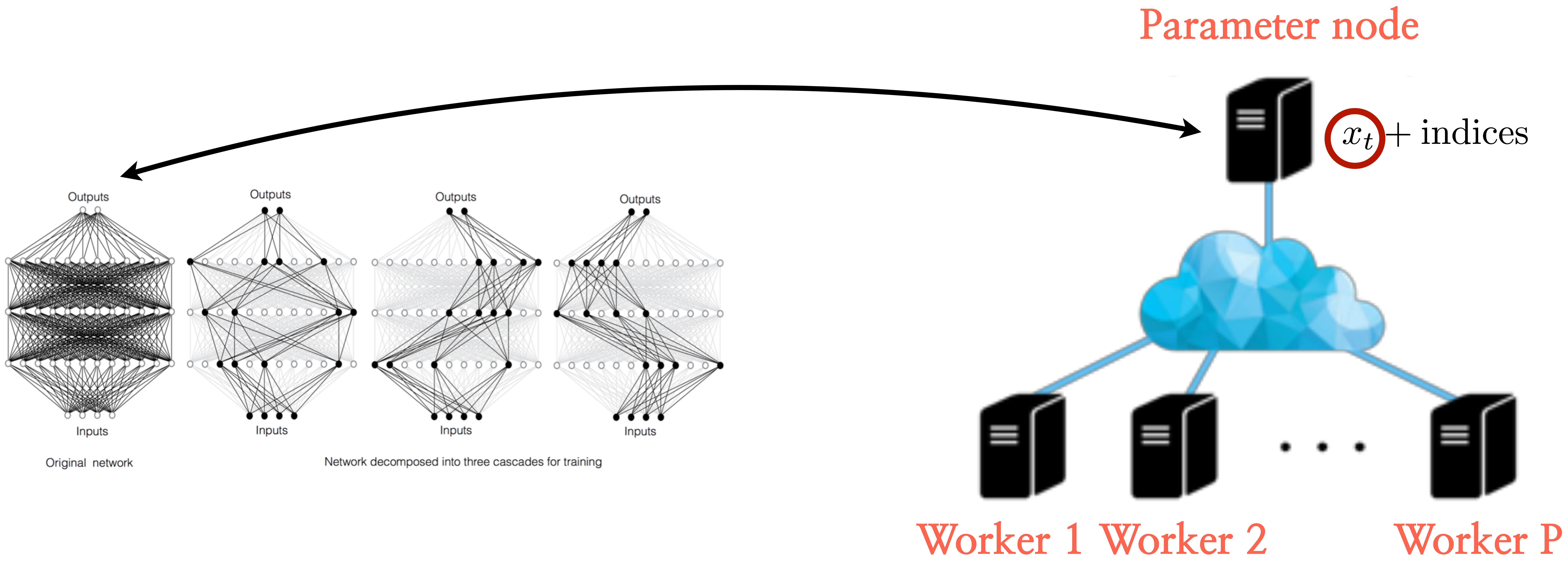
IST: Independent Subnet Training



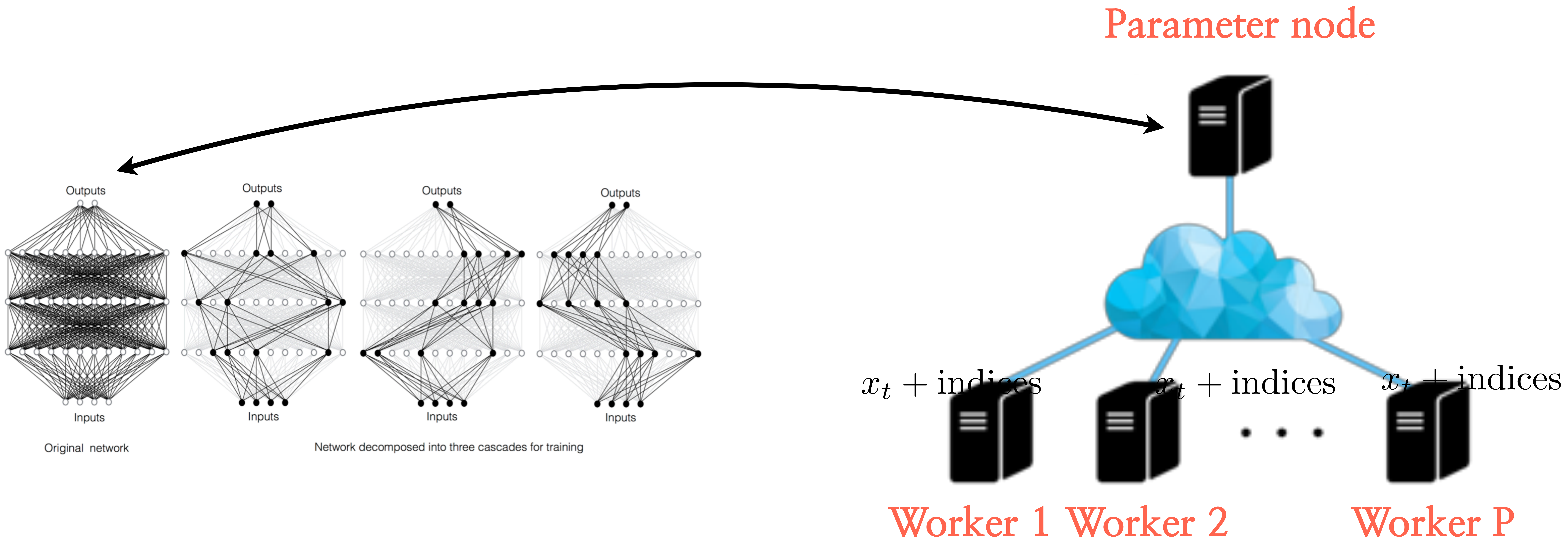
IST: Independent Subnet Training



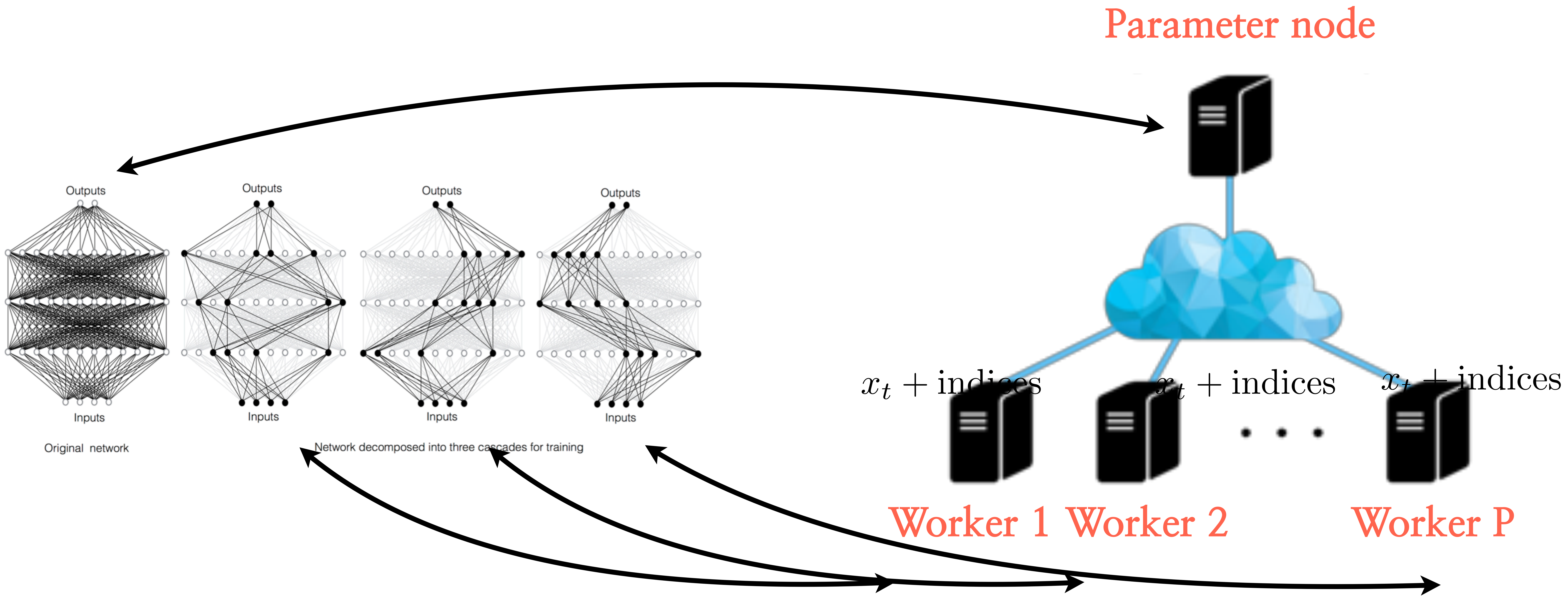
IST: Independent Subnet Training



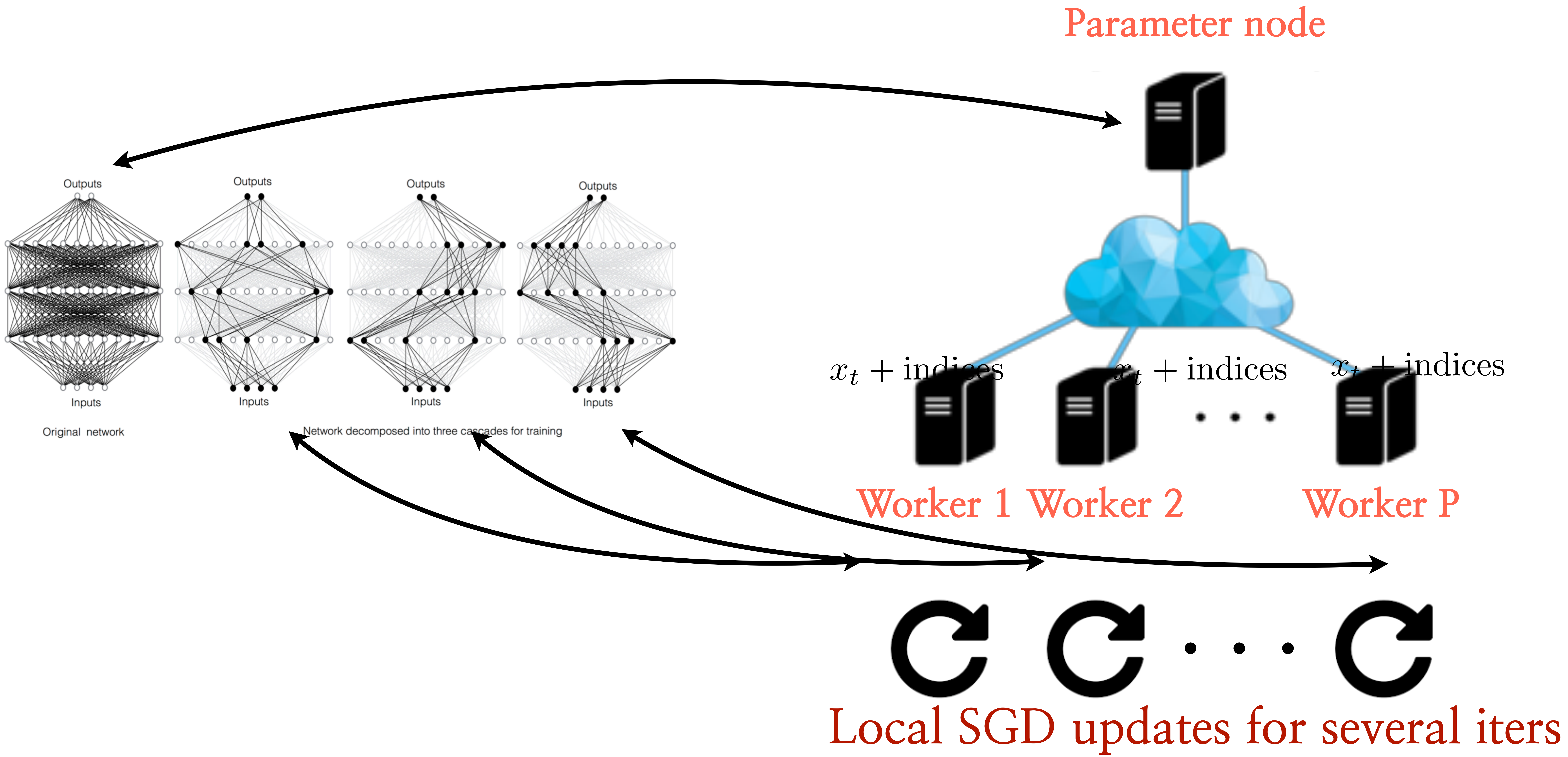
IST: Independent Subnet Training



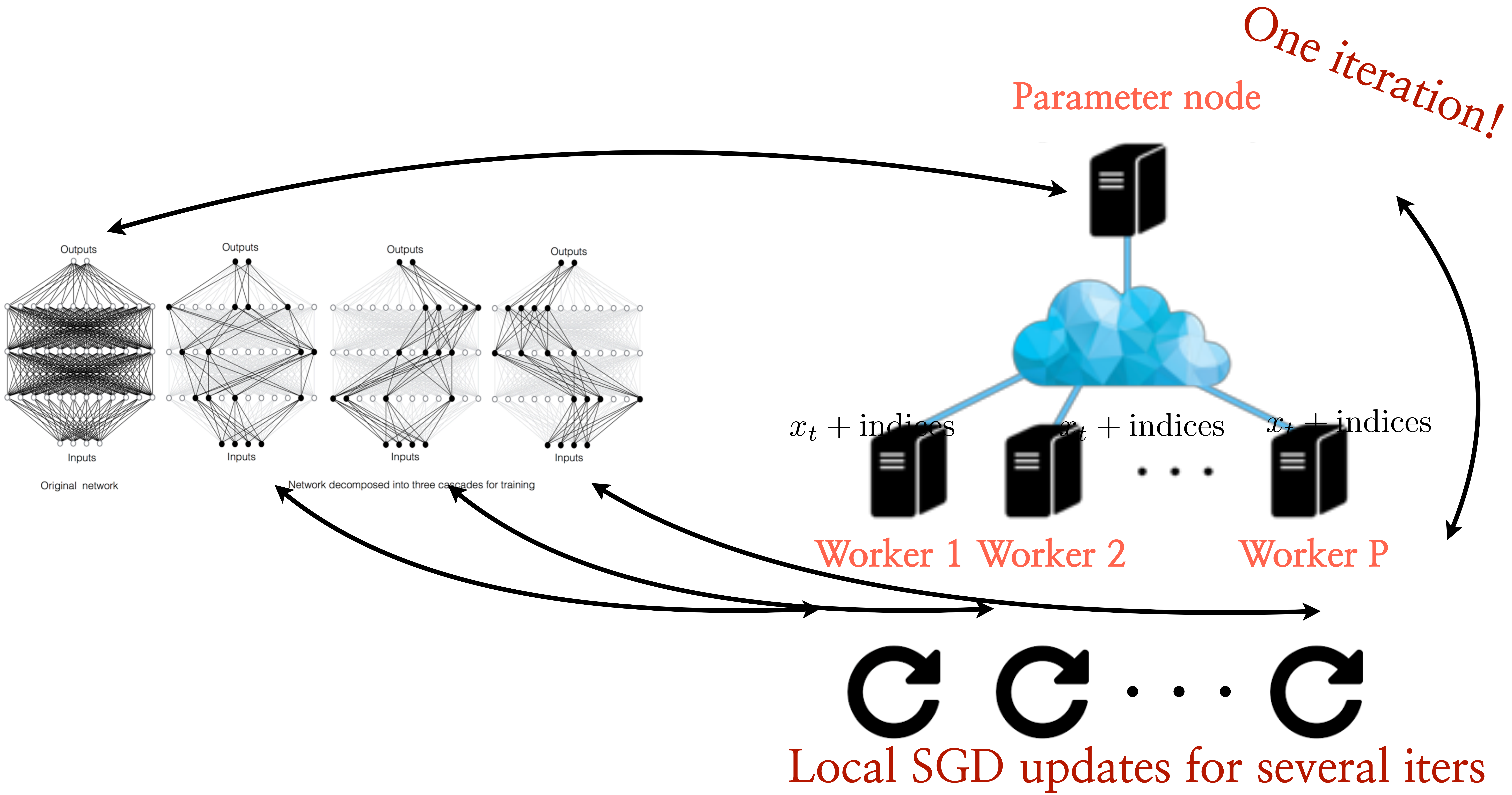
IST: Independent Subnet Training



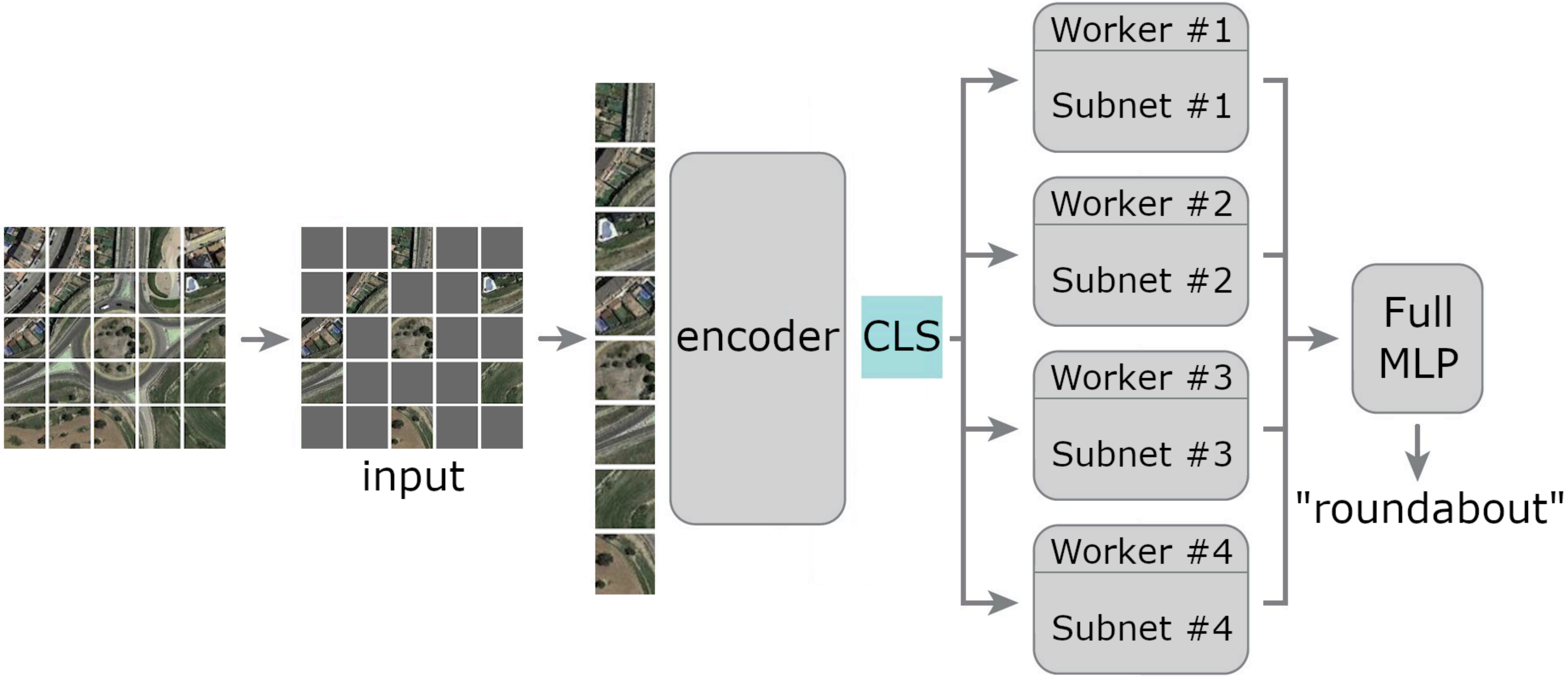
IST: Independent Subnet Training



IST: Independent Subnet Training

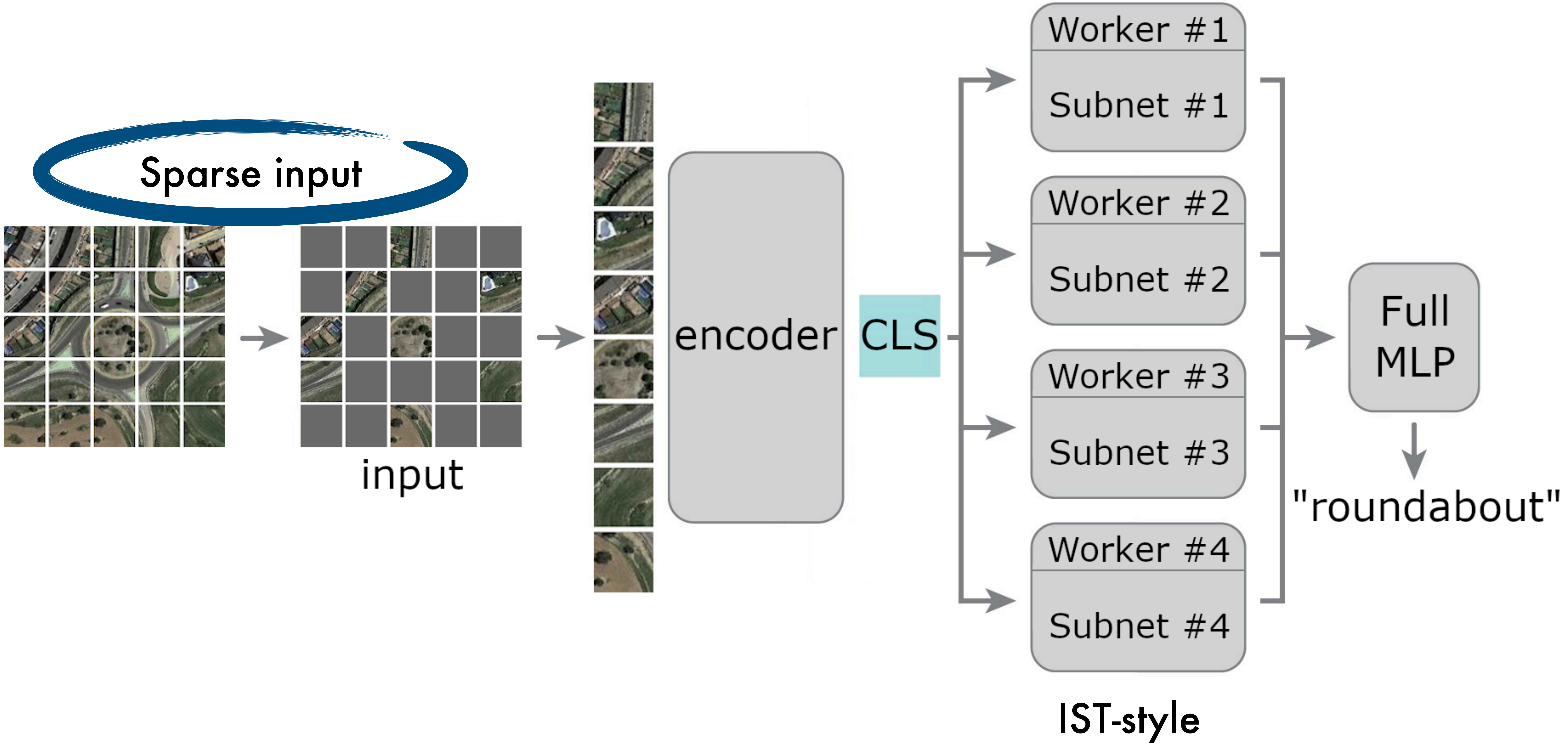


How does the model look like so far?

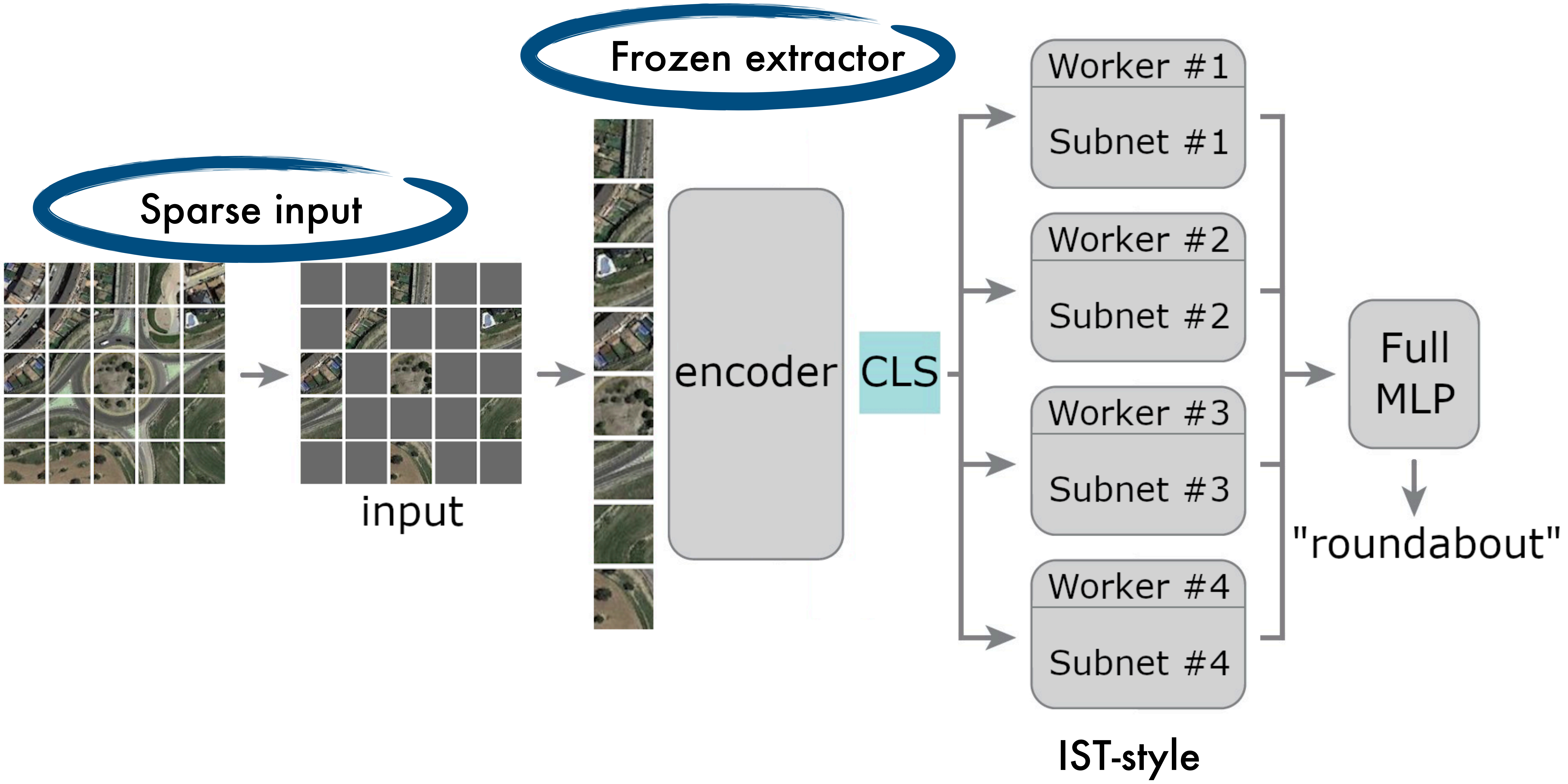


IST-style

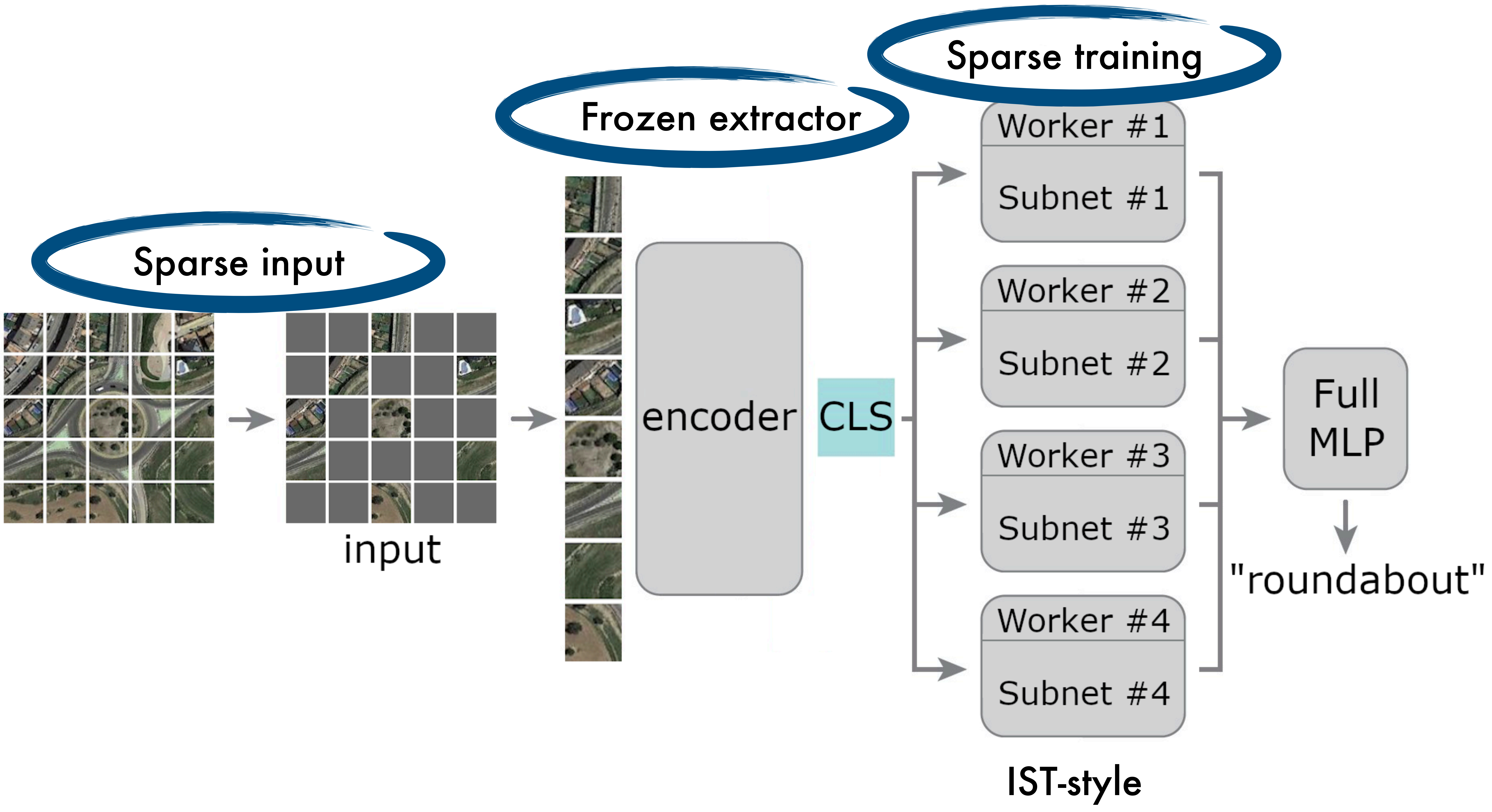
How does the model look like so far?



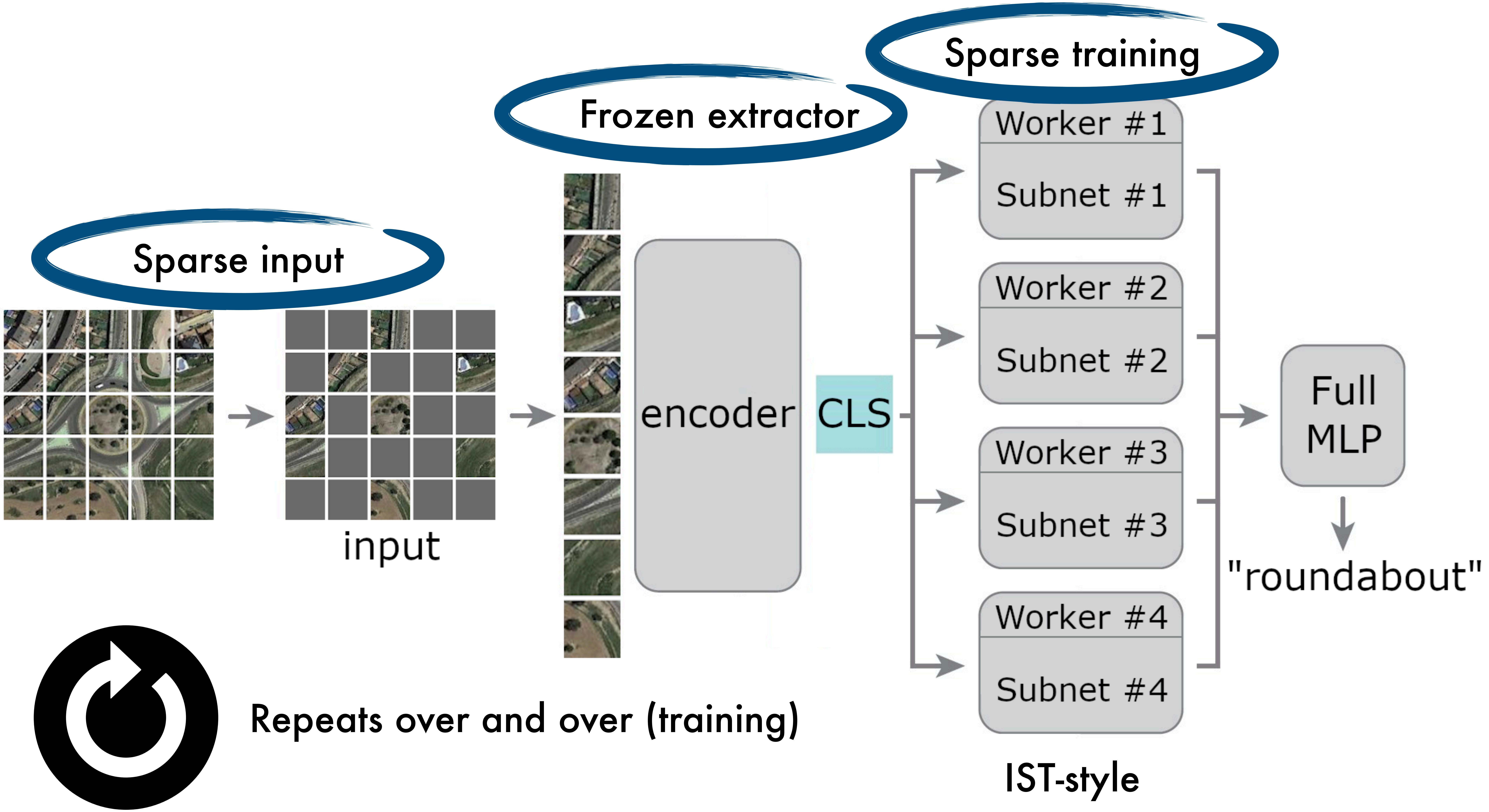
How does the model look like so far?



How does the model look like so far?



How does the model look like so far?



Repeats over and over (training)

Does it work?

- Simple tasks (Disclaimer: we need to do a better job on this moving forward)

Dataset	Classes	Image Size	Images per Class	Total (Training - Test Set)
CIFAR10	10	32×32	$6,000$	$60,000$ ($50,000$ - $10,000$)
RESISC45	45	256×256	700	$31,500$ ($27,000$ - $4,500$)
AID	30	600×600	$220 \sim 420$	$10,000$ ($8,500$ - $1,500$)

Does it work?

- Simple tasks (Disclaimer: we need to do a better job on this moving forward)

Dataset	Classes	Image Size	Images per Class	Total (Training - Test Set)
CIFAR10	10	32×32	$6,000$	$60,000$ ($50,000$ - $10,000$)
RESISC45	45	256×256	700	$31,500$ ($27,000$ - $4,500$)
AID	30	600×600	$220 \sim 420$	$10,000$ ($8,500$ - $1,500$)

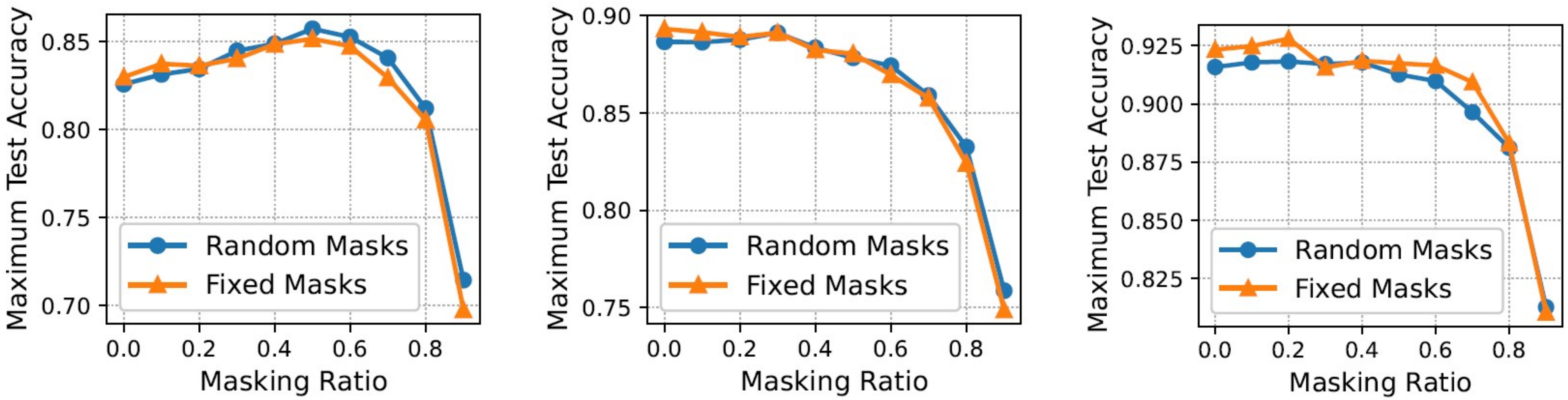
- Ablation study: *Can we use fixed masks to sparsity inputs?* Seems so!

Does it work?

- Simple tasks (Disclaimer: we need to do a better job on this moving forward)

Dataset	Classes	Image Size	Images per Class	Total (Training - Test Set)
CIFAR10	10	32×32	$6,000$	$60,000$ ($50,000$ - $10,000$)
RESISC45	45	256×256	700	$31,500$ ($27,000$ - $4,500$)
AID	30	600×600	$220 \sim 420$	$10,000$ ($8,500$ - $1,500$)

- Ablation study: *Can we use fixed masks to sparsity inputs?* Seems so!



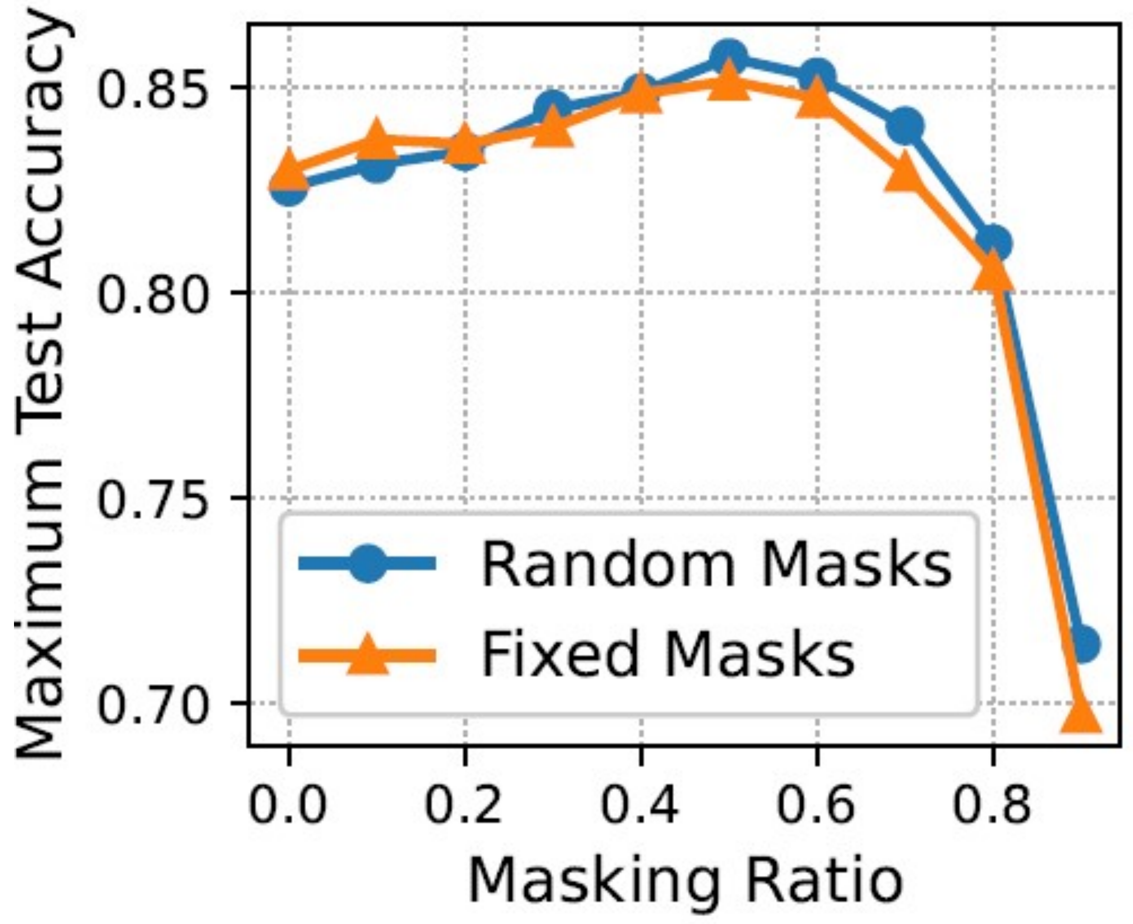
(a) CIFAR10 (b) RESISC45 (c) AID

Does it work?

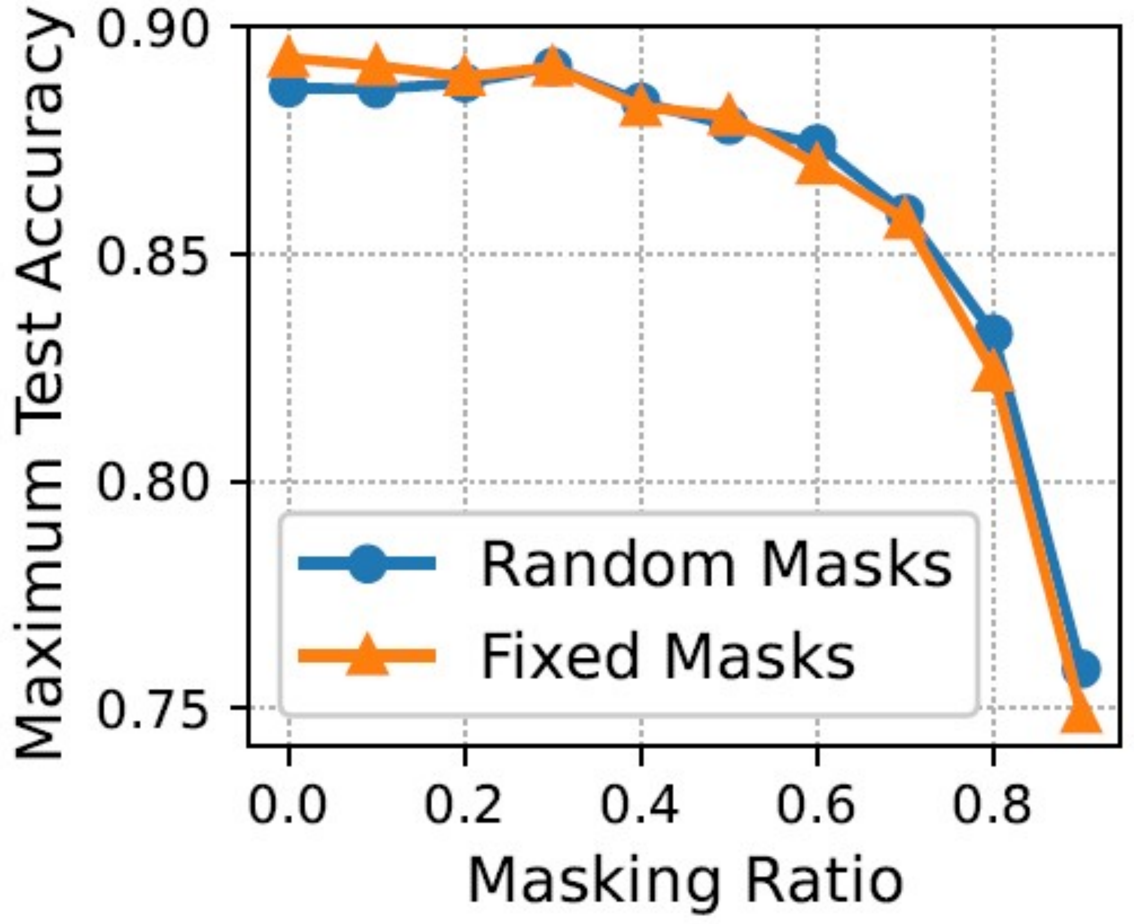
- Simple tasks (Disclaimer: we need to do a better job on this moving forward)

Dataset	Classes	Image Size	Images per Class	Total (Training - Test Set)
CIFAR10	10	32×32	6,000	60,000 (50,000 - 10,000)
RESISC45	45	256×256	700	31,500 (27,000 - 4,500)
AID	30	600×600	$220 \sim 420$	10,000 (8,500 - 1,500)

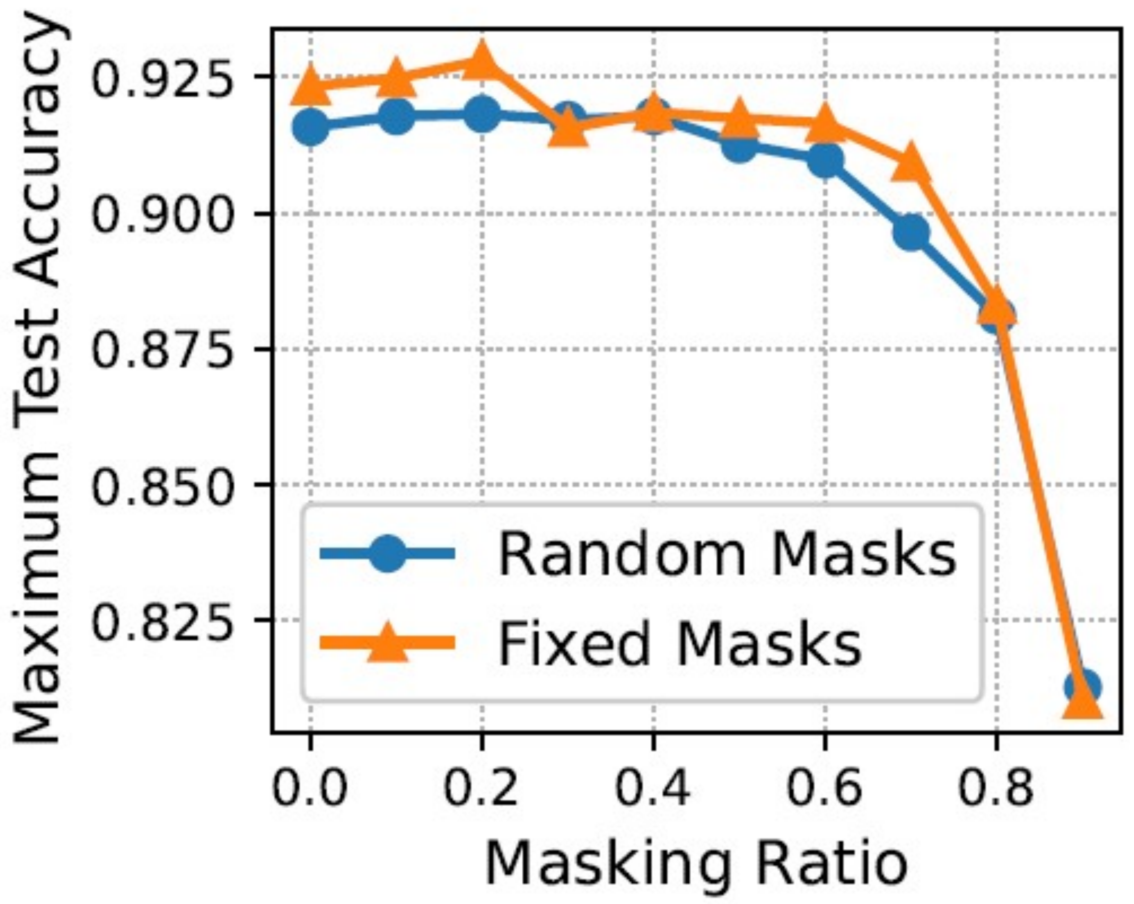
- Ablation study: *Can we use fixed masks to sparsity inputs?* Seems so!



(a) CIFAR10



(b) RESISC45



(c) AID

This means that:

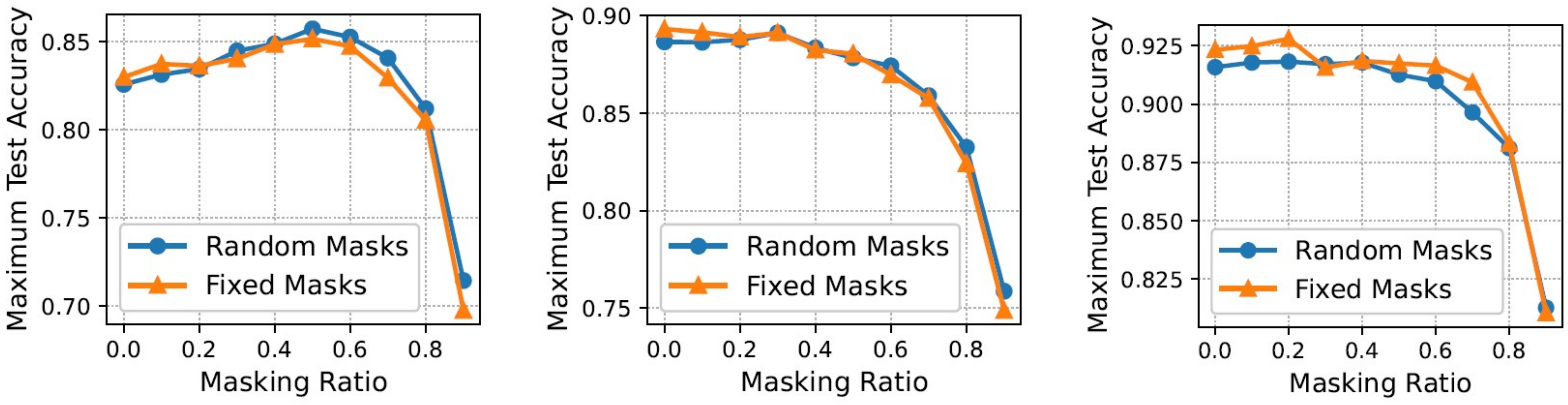
- Applications with natural sparsity could be solved fine (given enough data points)

Does it work?

- Simple tasks (Disclaimer: we need to do a better job on this moving forward)

Dataset	Classes	Image Size	Images per Class	Total (Training - Test Set)
CIFAR10	10	32×32	$6,000$	$60,000$ ($50,000$ - $10,000$)
RESISC45	45	256×256	700	$31,500$ ($27,000$ - $4,500$)
AID	30	600×600	$220 \sim 420$	$10,000$ ($8,500$ - $1,500$)

- Ablation study: *Can we use fixed masks to sparsity inputs?* Seems so!



(a) CIFAR10 (b) RESISC45 (c) AID

This means that:

- Applications with artificial input sparsity do not require random mask generation per iteration

Does it work?

- The system works for various masking ratios, preserving accuracy

Does it work?

- The system works for various masking ratios, preserving accuracy

	Masking Ratio	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CIFAR10	Masked Images Size (in GB)	0.205	0.184	0.164	0.143	0.123	0.102	0.082	0.061	0.041	0.020
	CLS Tokens Size (in GB)	0.1536									
	Max Accuracy	0.823	0.837	0.836	0.840	**0.848**	**0.851**	0.847	0.829	0.805	0.697
RESISC45	Masked Images Size (in GB)	7.078	6.370	5.662	4.954	4.247	3.539	2.831	2.123	1.416	0.708
	CLS Tokens Size (in GB)	0.083									
	Max Accuracy	**0.893**	**0.891**	0.889	**0.891**	0.882	0.880	0.869	0.857	0.824	0.749
AID	Masked Images Size (in GB)	12.240	11.016	9.792	8.568	7.344	6.120	4.896	3.672	2.448	1.224
	CLS Tokens Size (in GB)	0.026									
	Max Accuracy	0.923	**0.925**	**0.928**	0.916	0.918	0.917	0.916	0.909	0.883	0.810

Does it work?

- The system works for various masking ratios, preserving accuracy

More sparse input!

	Masking Ratio	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CIFAR10	Masked Images Size (in GB)	0.205	0.184	0.164	0.143	0.123	0.102	0.082	0.061	0.041	0.020
	CLS Tokens Size (in GB)	0.1536									
	Max Accuracy	0.823	0.837	0.836	0.840	**0.848**	**0.851**	0.847	0.829	0.805	0.697
RESISC45	Masked Images Size (in GB)	7.078	6.370	5.662	4.954	4.247	3.539	2.831	2.123	1.416	0.708
	CLS Tokens Size (in GB)	0.083									
	Max Accuracy	**0.893**	**0.891**	0.889	**0.891**	0.882	0.880	0.869	0.857	0.824	0.749
AID	Masked Images Size (in GB)	12.240	11.016	9.792	8.568	7.344	6.120	4.896	3.672	2.448	1.224
	CLS Tokens Size (in GB)	0.026									
	Max Accuracy	0.923	**0.925**	**0.928**	0.916	0.918	0.917	0.916	0.909	0.883	0.810

Does it work?

- The system works for various masking ratios, preserving accuracy

More sparse input!

Accuracy preserved!

	Masking Ratio	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CIFAR10	Masked Images Size (in GB)	0.205	0.184	0.164	0.143	0.123	0.102	0.082	0.061	0.041	0.020
	CLS Tokens Size (in GB)					0.1536					
	Max Accuracy	0.823	0.837	0.836	0.840	**0.848**	**0.851**	0.847	0.829	0.805	0.697
RESISC45	Masked Images Size (in GB)	7.078	6.370	5.662	4.954	4.247	3.539	2.831	2.123	1.416	0.708
	CLS Tokens Size (in GB)					0.083					
	Max Accuracy	**0.893**	**0.891**	0.889	**0.891**	0.882	0.880	0.869	0.857	0.824	0.749
AID	Masked Images Size (in GB)	12.240	11.016	9.792	8.568	7.344	6.120	4.896	3.672	2.448	1.224
	CLS Tokens Size (in GB)					0.026					
	Max Accuracy	0.923	**0.925**	**0.928**	0.916	0.918	0.917	0.916	0.909	0.883	0.810

Does it work?

- The system works for various masking ratios, preserving accuracy

More sparse input!

	Masking Ratio	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CIFAR10	Masked Images Size (in GB)	0.205	0.184	0.164	0.143	0.123	0.102	0.082	0.061	0.041	0.020
	CLS Tokens Size (in GB)	0.1536									
	Max Accuracy	0.823	0.837	0.836	0.840	**0.848**	**0.851**	0.847	0.829	0.805	0.697
RESISC45	Masked Images Size (in GB)	7.078	6.370	5.662	4.954	4.247	3.539	2.831	2.123	1.416	0.708
	CLS Tokens Size (in GB)	0.083									
	Max Accuracy	**0.893**	**0.891**	0.889	**0.891**	0.882	0.880	0.869	0.857	0.824	0.749
AID	Masked Images Size (in GB)	12.240	11.016	9.792	8.568	7.344	6.120	4.896	3.672	2.448	1.224
	CLS Tokens Size (in GB)	0.026									
	Max Accuracy	0.923	**0.925**	**0.928**	0.916	0.918	0.917	0.916	0.909	0.883	0.810

Does it work?

- The system works for various masking ratios, preserving accuracy

More sparse input!

	Masking Ratio	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CIFAR10	Masked Images Size (in GB)	0.205	0.184	0.164	0.143	0.123	0.102	0.082	0.061	0.041	0.020
	CLS Tokens Size (in GB)	0.1536									
	Max Accuracy	0.823	0.837	0.836	0.840	**0.848**	**0.851**	0.847	0.829	0.805	0.697
RESISC45	Masked Images Size (in GB)	7.078	6.370	5.662	4.954	4.247	3.539	2.831	2.123	1.416	0.708
	CLS Tokens Size (in GB)	0.083									
	Max Accuracy	**0.893**	**0.891**	0.889	**0.891**	0.882	0.880	0.869	0.857	0.824	0.749
AID	Masked Images Size (in GB)	12.240	11.016	9.792	8.568	7.344	6.120	4.896	3.672	2.448	1.224
	CLS Tokens Size (in GB)	0.026									
	Max Accuracy	0.923	**0.925**	**0.928**	0.916	0.918	0.917	0.916	0.909	0.883	0.810

Accuracy even improved!

Final comments

- This work is far from complete; some next steps:

Final comments

- This work is far from complete; some next steps:
  - More datasets (and more challenging ones) should be considered

Final comments

- This work is far from complete; some next steps:
  - More datasets (and more challenging ones) should be considered
  - More applications (within CV and beyond, like NLP) should be considered
    - Especially, in the BERT-type of models, input sparsity = token compression (important)!

Final comments

- This work is far from complete; some next steps:
  - More datasets (and more challenging ones) should be considered
  - More applications (within CV and beyond, like NLP) should be considered
    - Especially, in the BERT-type of models, input sparsity = token compression (important)!
  - We have been “cheating” with using pretrained model (too large for some applications)
    - Is it possible to have pure end-2-end sparse training? What about the output layer?

Final comments

- This work is far from complete; some next steps:
  - More datasets (and more challenging ones) should be considered
  - More applications (within CV and beyond, like NLP) should be considered
    - Especially, in the BERT-type of models, input sparsity = token compression (important)!
  - We have been “cheating” with using pretrained model (too large for some applications)
    - Is it possible to have pure end-2-end sparse training? What about the output layer?
    - Recent work by Rice’s group that goes beyond this work:
      - MoEs for sparse training: FedJets [2023], MoPs [2023], … (collaboration with MSR)
      - Lottery ticket hypothesis extensions: Strong LTH [2023], How much pretraining [2023]

Final comments

- This work is far from complete; some next steps:
  - More datasets (and more challenging ones) should be considered
  - More applications (within CV and beyond, like NLP) should be considered
    - Especially, in the BERT-type of models, input sparsity = token compression (important)!
  - We have been "cheating" with using pretrained model (too large for some applications)
    - Is it possible to have pure end-2-end sparse training? What about the output layer?
    - Recent work by Rice's group that goes beyond this work:
      - MoEs for sparse training: FedJets [2023], MoPs [2023], … (collaboration with MSR)
      - Lottery ticket hypothesis extensions: Strong LTH [2023], How much pretraining [2023]
  - Can we theoretically characterize the input layer sparsity?
    - Even an NTK analysis would be a great start

I would love to be there :(- Thank you Daniel!

For questions: anastasios@rice.edu