Meta-Learning Sparse Implicit Neural Representations

Jaeho Lee*[1], Jihoon Tack*[1], Namhoon Lee[2], Jinwoo Shin[1]

Korea Advanced Institute of Science and Technology (KAIST)[1]

Ulsan National Institute of Science and Technology (UNIST)[2]

Korean Conference on Computer Vision (KCCV) 2022

Conference on Neural Information Processing Systems (NeurIPS) 2021

* equal contribution

Implicit Neural Representations

INR (Implicit Neural Representation) is an emerging paradigm

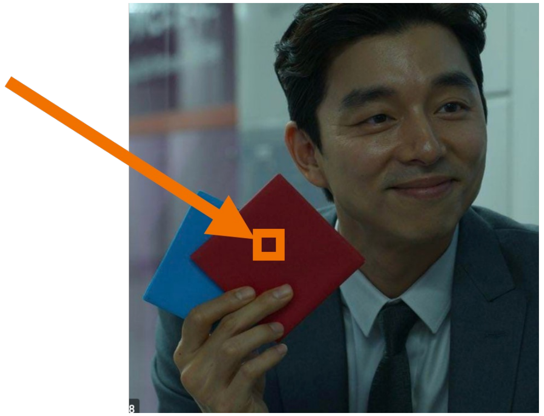
that uses neural nets for **modeling individual data** (instead of for, e.g., predictions)

Implicit Neural Representations

INR (Implicit Neural Representation) is an emerging paradigm that uses neural nets for **modeling individual data** (instead of for, e.g., predictions)

For instance, an **RGB image** can be represented as a function taking a form

$$\begin{array}{ccc} \texttt{coords} & \longrightarrow & \texttt{values} \\ (X, Y) & \longrightarrow & (R, G, B) \\ \mathbb{R}^2 & \longrightarrow & \mathbb{R}^3 \end{array}$$



$$f_{\text{img}}\left(\frac{76}{255}, \frac{152}{255}\right) = (0.95, 0.03, 0.04)$$

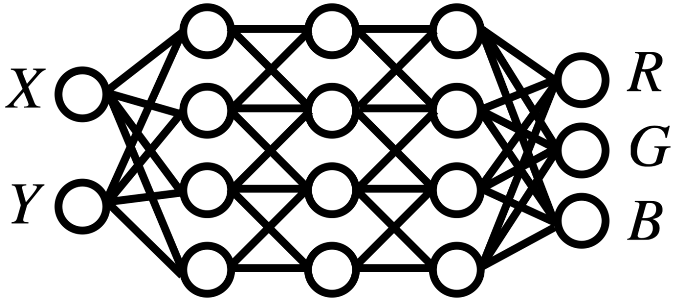
Implicit Neural Representations

INR (Implicit Neural Representation) is an emerging paradigm that uses neural nets for **modeling individual data** (instead of for, e.g., predictions)

For instance, an **RGB image** can be represented as a function taking a form

$$\begin{array}{ccc} \texttt{coords} & \longrightarrow & \texttt{values} \\ (\mathrm{X},\mathrm{Y}) & \longrightarrow & (\mathrm{R},\mathrm{G},\mathrm{B}) \\ \mathbb{R}^2 & \longrightarrow & \mathbb{R}^3 \end{array}$$

This can be learned by a neural network with $d_{\text{in}} = 2, d_{\text{out}} = 3$



$$\min_{\theta} \left\| f_{\text{img}} - f_{\theta} \right\|_{L_2}$$

Why is INR interesting?

INR has the **potential** to be a popular form of **data representation** in a near future!

Why is INR interesting?

INR has the **potential** to be a popular form of **data representation** in a near future!

- Interesting point 1. Effective at novel view synthesis (or resolution free!) [1,2]

Pixels Bilinear INR-decoder

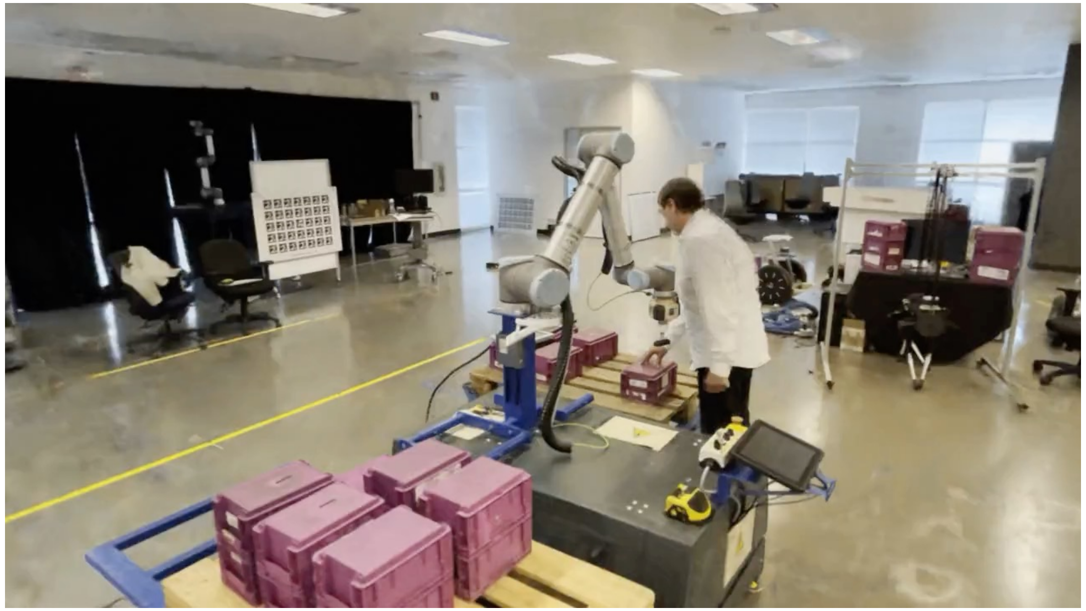
Rendering: Image → 3D

[1] Chen et al. Learning Continuous Image Representation with Local Implicit Image Function. CVPR 2021
[2] Mildenhall et al. NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images. CVPR 2022

Why is INR interesting?

INR has the **potential** to be a popular form of **data representation** in a near future!

- Interesting point 2. Represent complex signals [1,2], e.g., large-scale 3d scenes, videos



[1] Tancik et al. Block-NeRF: Scalable Large Scene Neural View Synthesis. CVPR 2022
[2] Muller et al. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. SIGGRAPH 2022

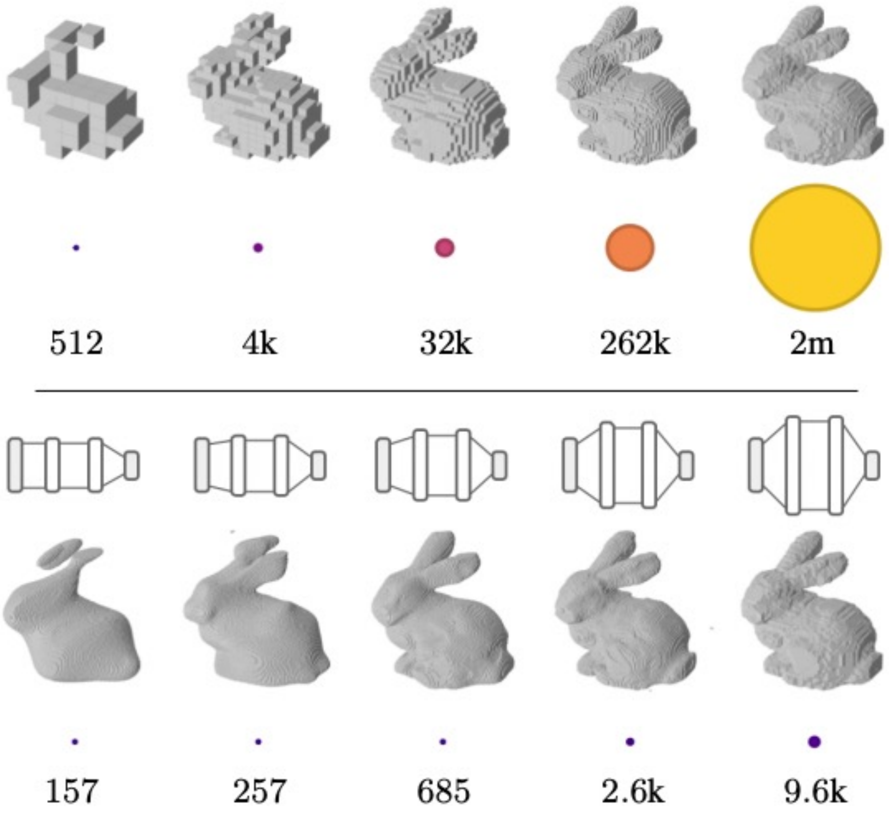
Why is INR interesting?

INR has the **potential** to be a popular form of **data representation** in a near future!

- Interesting point 3. Scalable (in terms of memory) [1]

Voxels →

INRs →

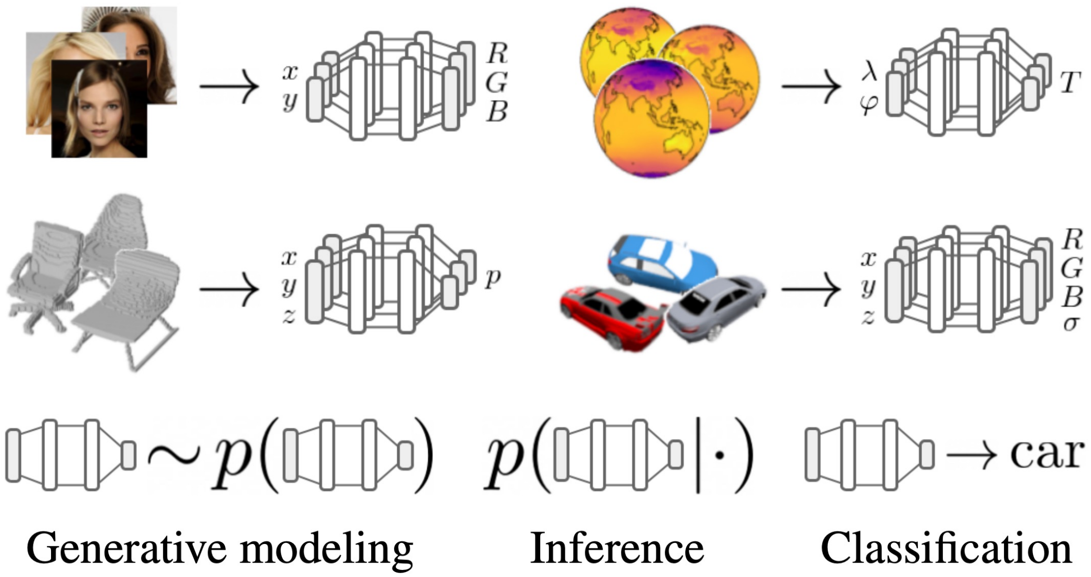


[1] Dupont et al. From data to functa: Your data point is a function and you should treat it like one. ICML 2022

Why is INR interesting?

INR has the **potential** to be a popular form of **data representation** in a near future!

- Interesting point 4. INR it-self can be used as a data point!



Step 1: Fit data points into INRs

Step 2: Run downstream tasks, e.g., classification

[1] Dupont et al. From data to functa: Your data point is a function and you should treat it like one. ICML 2022

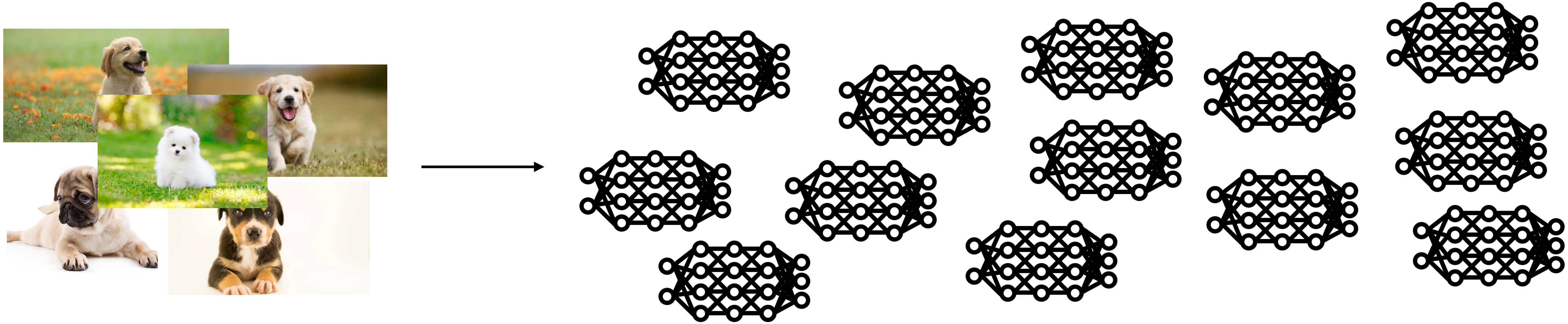
An Obstacle

Q. Can we scale such an idea to a **Big Dataset**?

An Obstacle

Q. Can we scale such an idea to a **Big Dataset**?

A. No—at least for now—because of the cost to **train & store** all those models.



Imagine training ImageNet number of these....

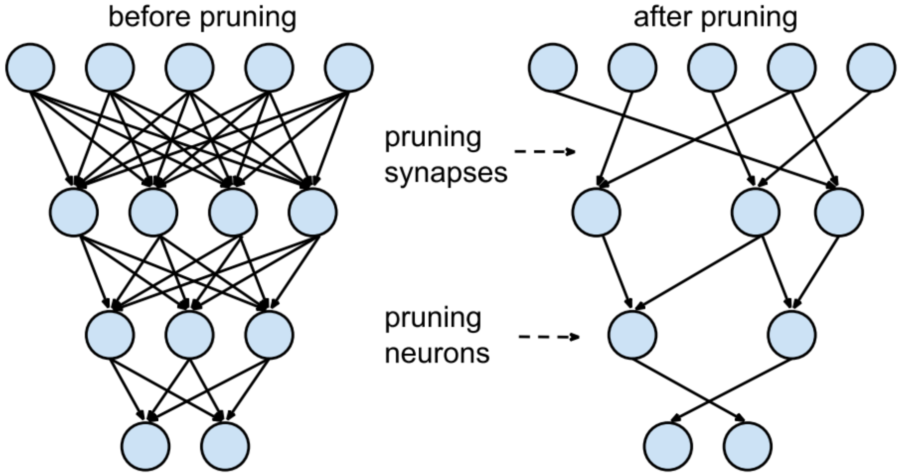
Naïve Way

Wait... but we have many **model compression** techniques, right?

Naïve Way

Wait... but we have many **model compression** techniques, right?

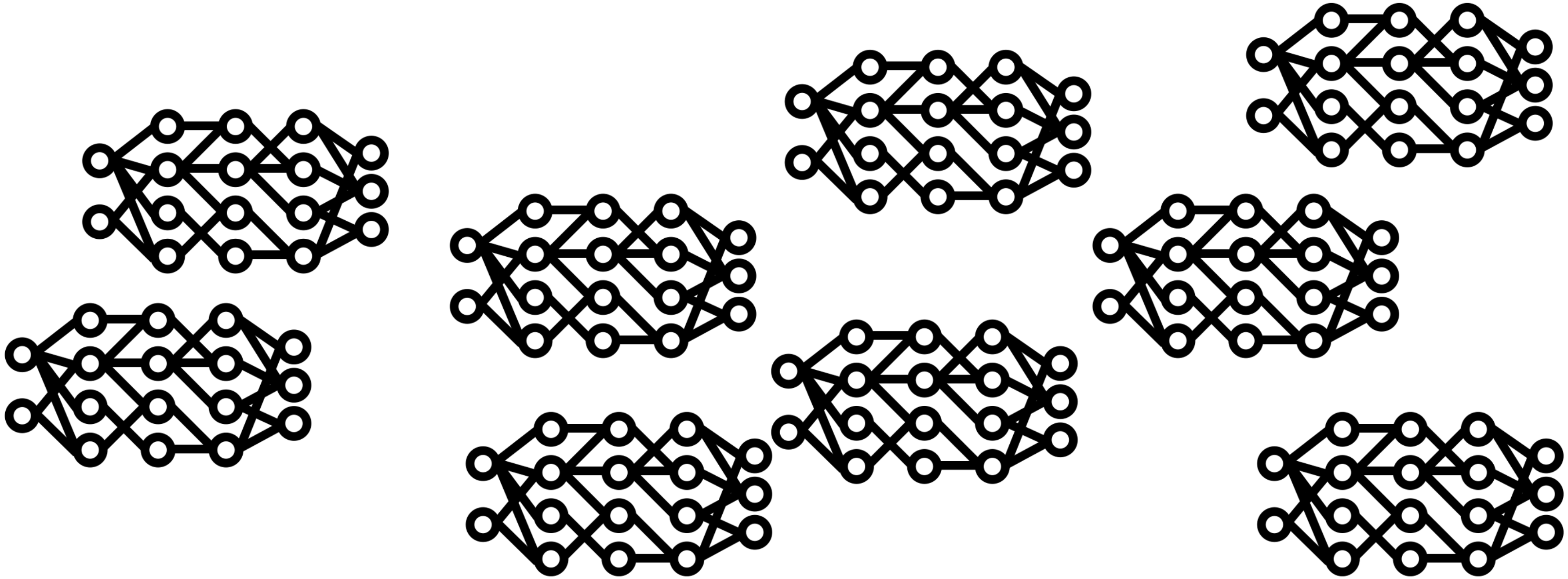
- What if we make **pruned** versions of INRs?



Naïve Way

Wait... but we have many **model compression** techniques, right?

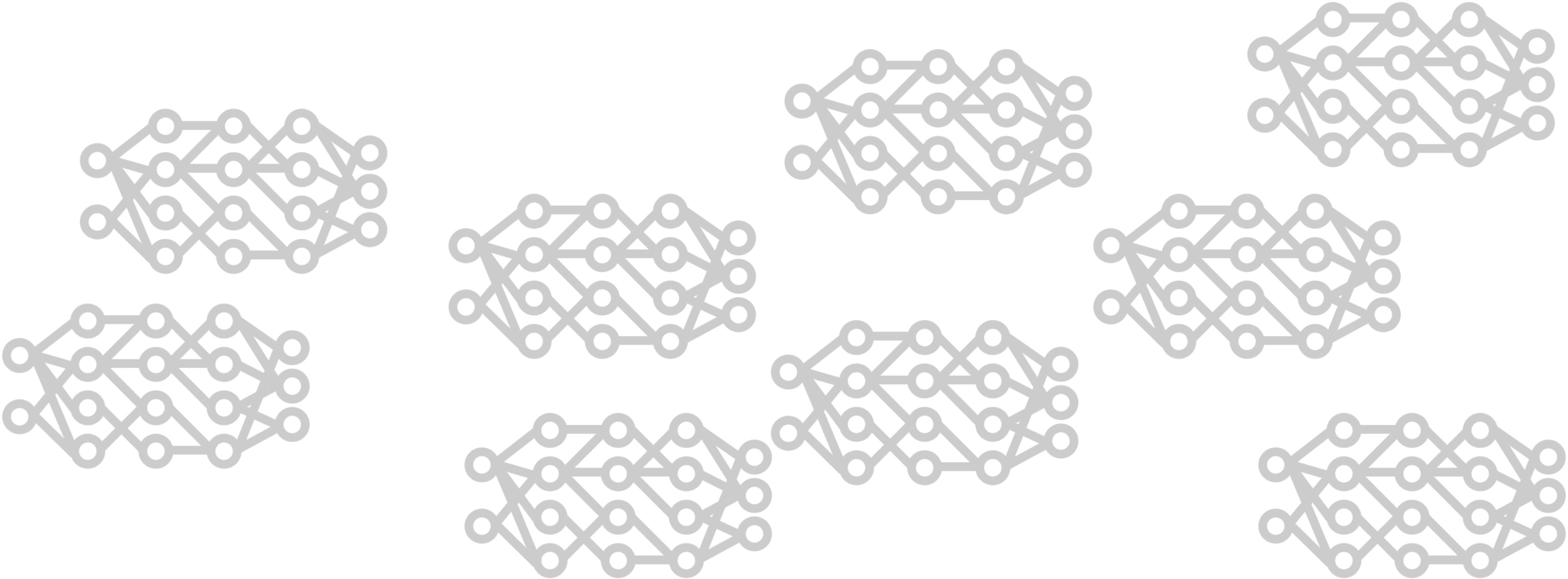
- What if we make **pruned** versions of INRs?



Naïve Way

Wait... but we have many **model compression** techniques, right?

- What if we make **pruned** versions of INRs?



- But generating sparse models takes a **much longer training time!! (up to 10x)**
  (as we gradually prune them over extended training time)

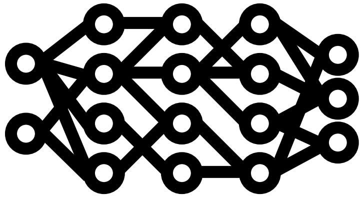
A Better Way

Can we make a **sparse initial model**?

Model with less parameters

+

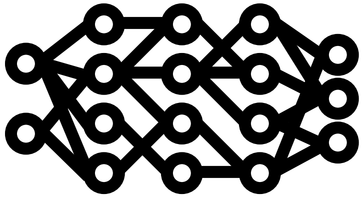
Good initalization



A Better Way

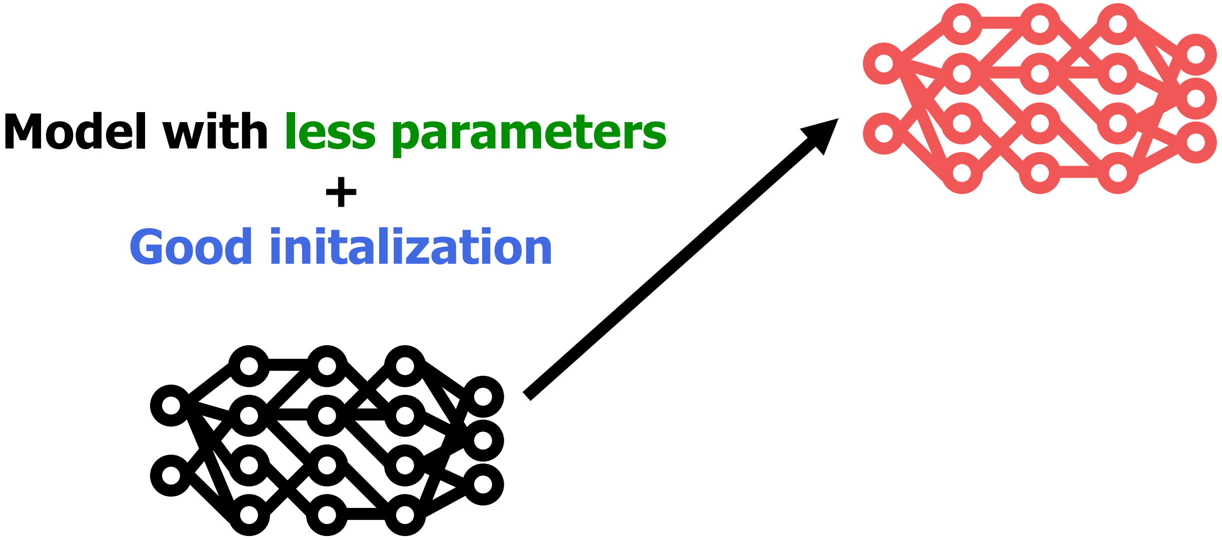
Can we make a **sparse initial model**, which can be **efficiently trained** to fit each signal?

Model with less parameters
+
Good initalization



A Better Way

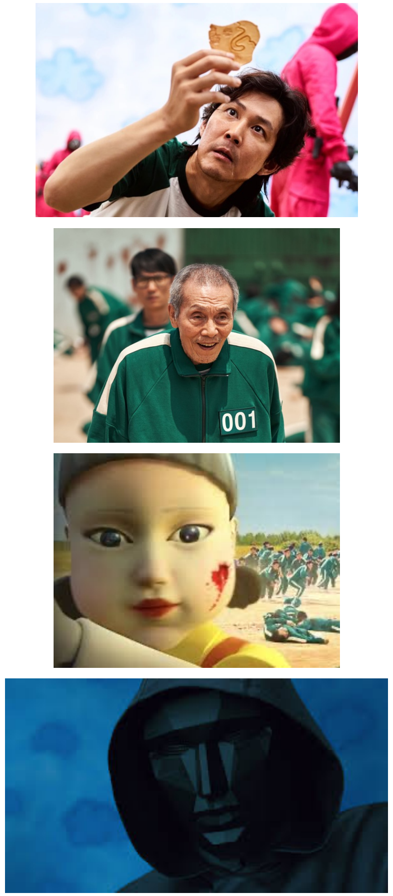
Can we make a **sparse initial model**, which can be **efficiently trained** to fit each signal?



Model with less parameters

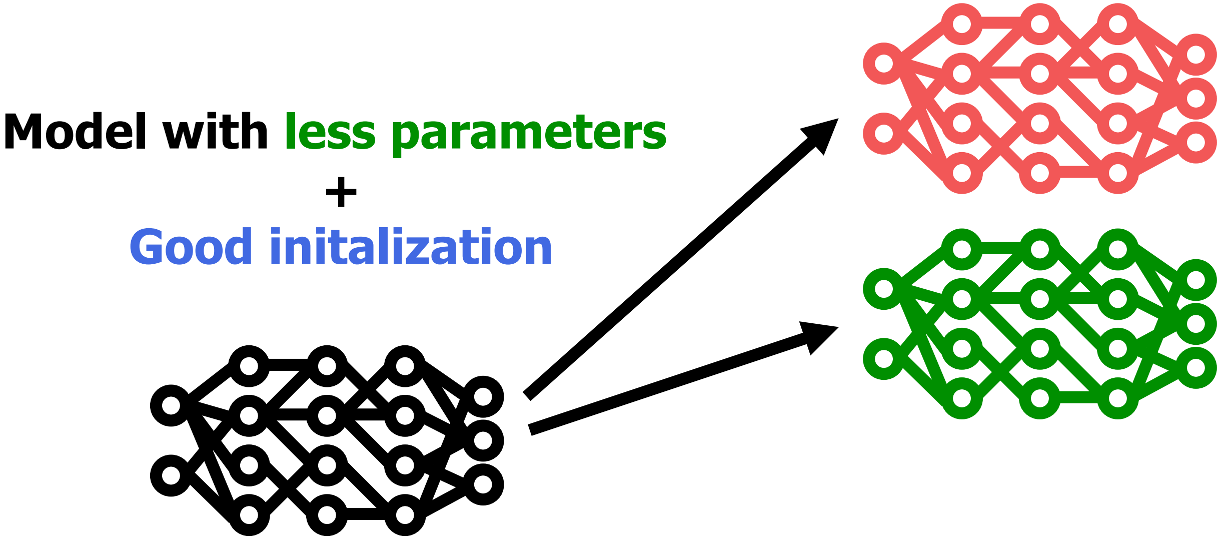
+

Good initalization



A Better Way

Can we make a **sparse initial model**, which can be **efficiently trained** to fit each signal?



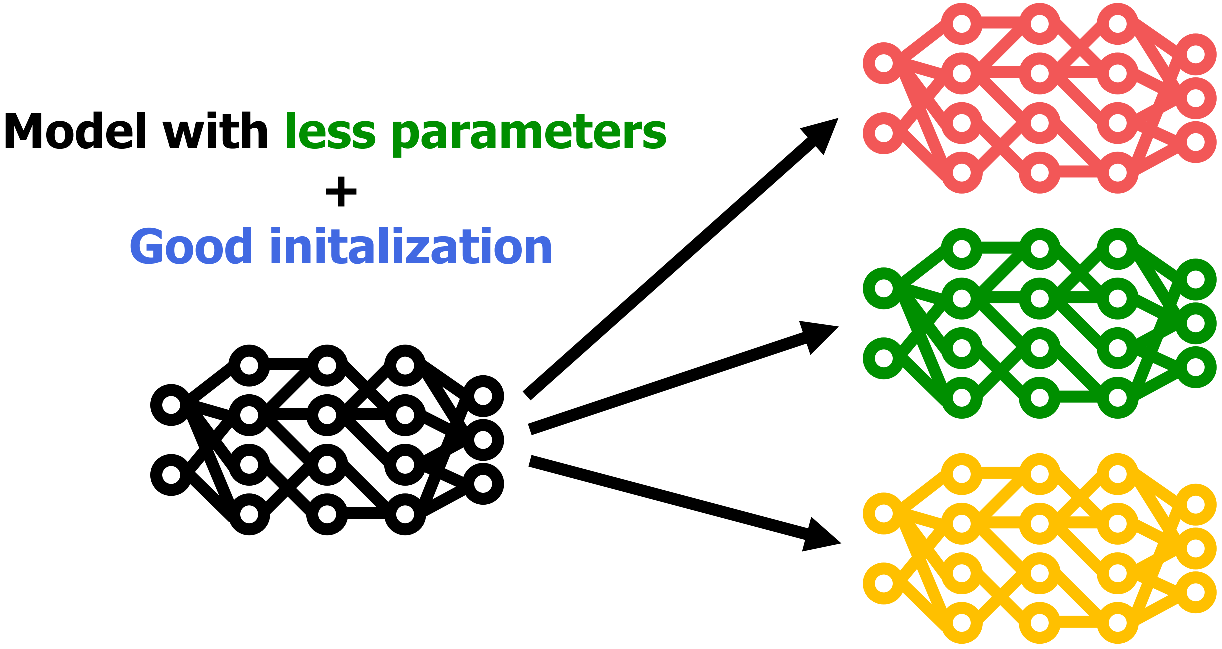
Model with less parameters

+

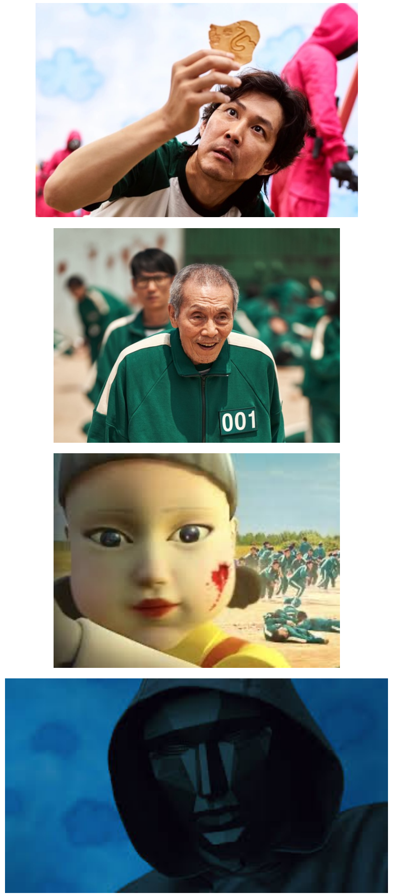
Good initalization

A Better Way

Can we make a **sparse initial model**, which can be **efficiently trained** to fit each signal?

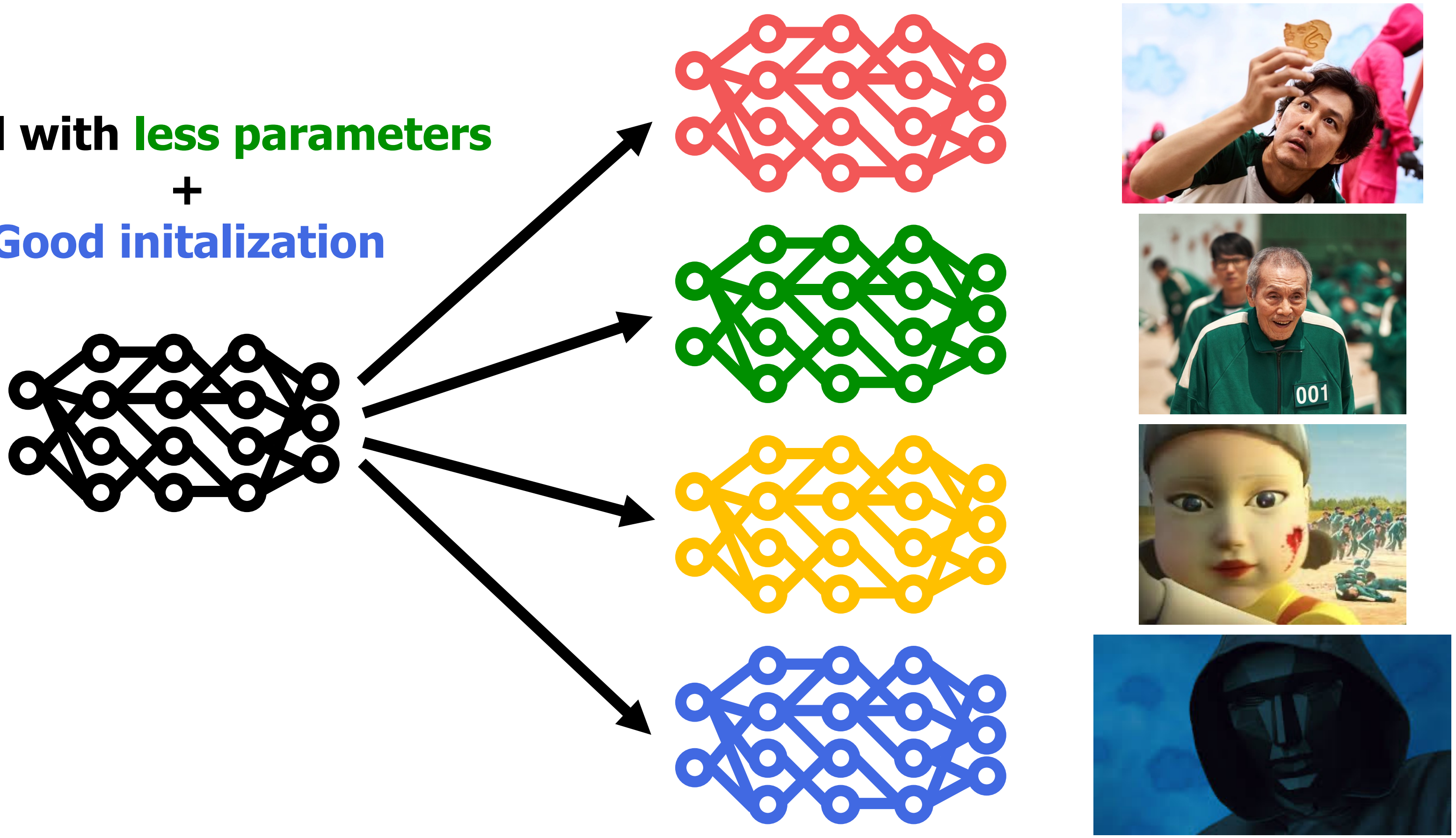


Model with less parameters
+
Good initalization



A Better Way

Can we make a **sparse initial model**, which can be **efficiently trained** to fit each signal?



Model with less parameters

+

Good initalization

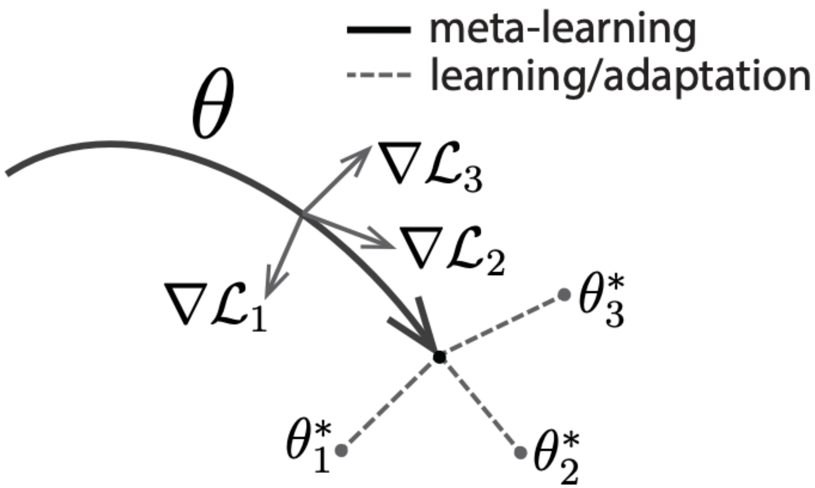
Method: Meta-SparseINR

How can we generate such a **sparse initial model**?

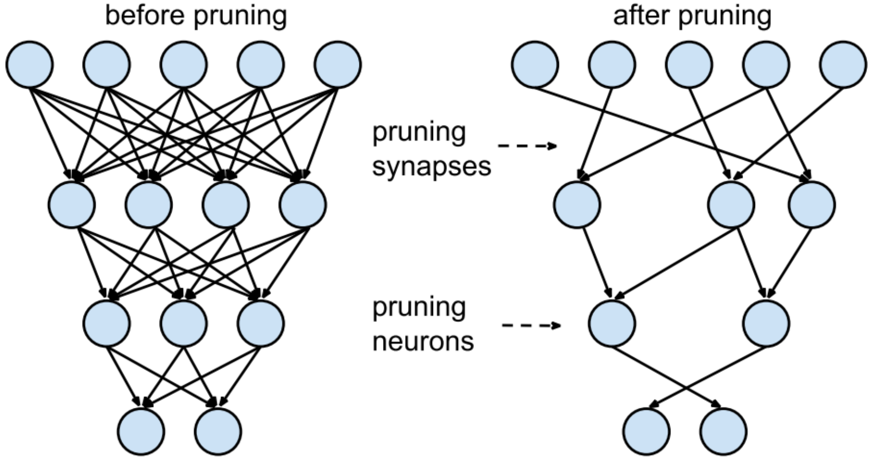
Method: Meta-SparseINR

How can we generate such a **sparse initial model**?

- Combine meta-learning and network pruning!



Meta-Learning

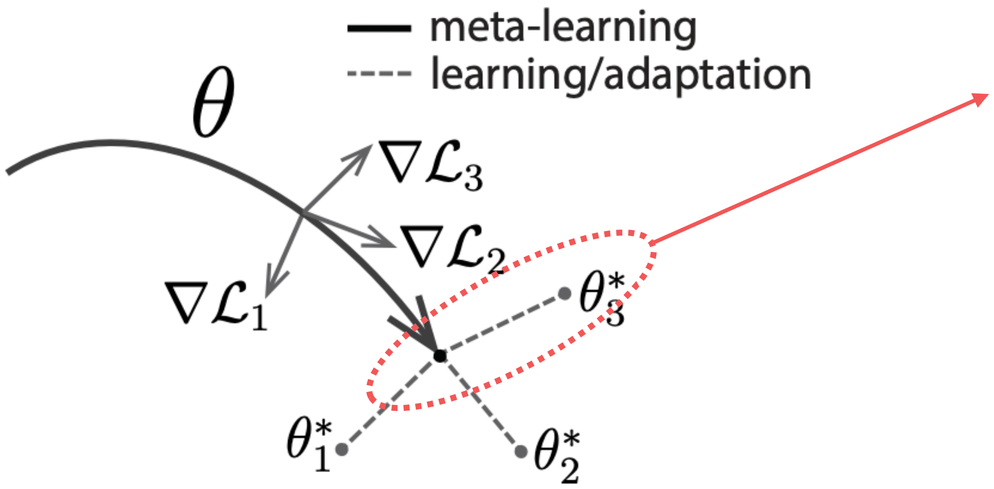


Network pruning

Model-Agnostic Meta-Learning (MAML) [1]

Learning **initializations** of a network that

- adapts fast with a small number of gradient steps
- can easily generalize to various model architecture and tasks



MAML

Objective of MAML on INRs?

Generalize on the **signal** after adapting with few-step gradients

$$\min_{\theta} \sum_{\mathcal{T}_i\sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}) = \sum_{\mathcal{T}_i\sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta-\alpha\nabla_{\theta}\mathcal{L}_{\mathcal{T}_i}(f_{\theta})}\right)$$

[1] Finn et al. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, ICML 2017

Meta-Learning INRs

For example, utilizing meta-learning on learning 3D shapes to efficiently adapt to new shapes [1]

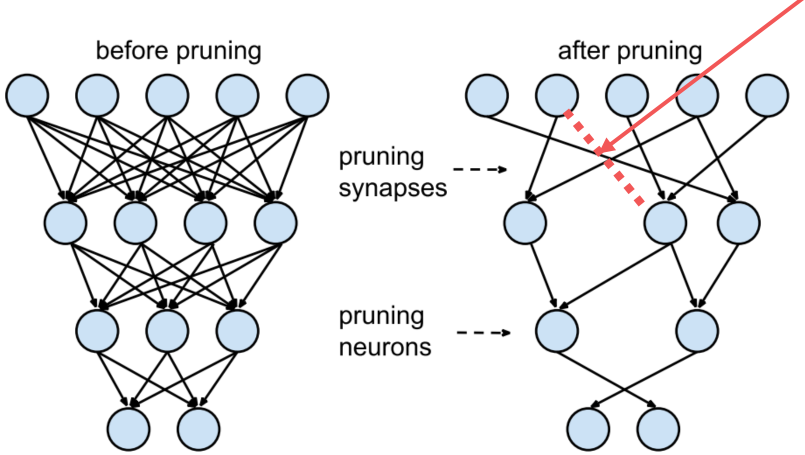
[1] Sitzmann et al. MetaSDF: Meta-Learning Signed Distance Functions, NeurIPS 2020

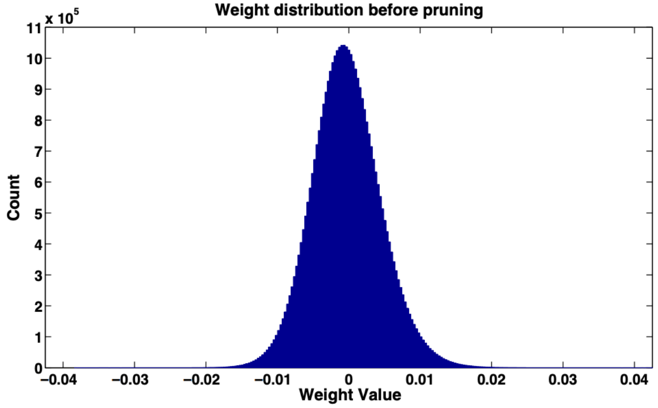
Magnitude Pruning (MP)

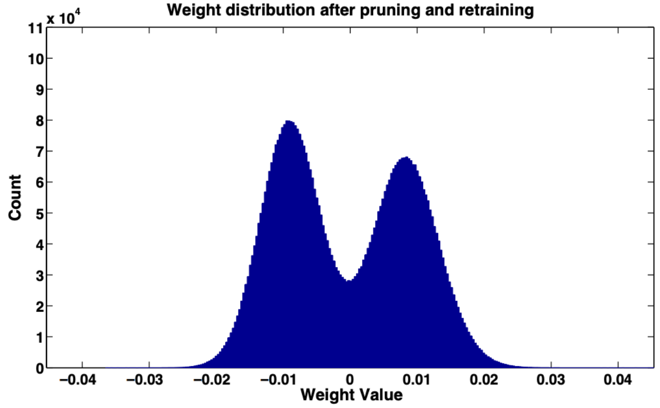
Which weight parameter to prune?

- Sort the weight values and **prune the parameters with small weights**!

small $|\theta_i|$







Although MP is a somewhat old technique, it is still a very effective tool to prune the network

[1] Learning both Weights and Connections for Efficient Neural Networks, Han et al., NeurIPS 2015

Method: Meta-SparseINR

How can we generate such **sparse initial model**?

Observation: Algorithms for pruning & efficient adaptation have one thing in common;

Network Weights play an essential role!

- **Pruning:** Removing edges with smallest weight magnitudes works surprisingly well!
- **Adaptation:** Can be done via gradient-based meta-learning (e.g., MAML) using weights.

Method: Meta-SparseINR

How can we generate such **sparse initial model**?

Observation: Algorithms for pruning & efficient adaptation have one thing in common;

Network Weights play an essential role!

- **Pruning:** Removing edges with smallest weight magnitudes works surprisingly well!
- **Adaptation:** Can be done via gradient-based meta-learning (e.g., MAML) using weights.

Idea: Meta-learned weights can be directly used as a pruning saliency score:

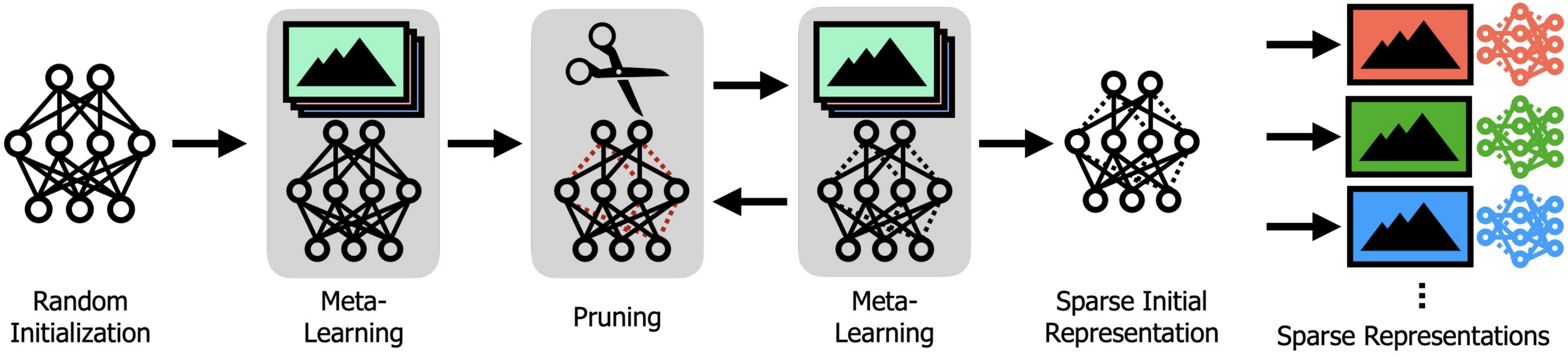
1. Meta-train a INR on a set of signals
2. Prune some connections based on meta-learned weights
3. Repeat

Method: Meta-SparseINR

How can we generate such **sparse initial model**?

Observa

- **Pruni**
- **Adapt**

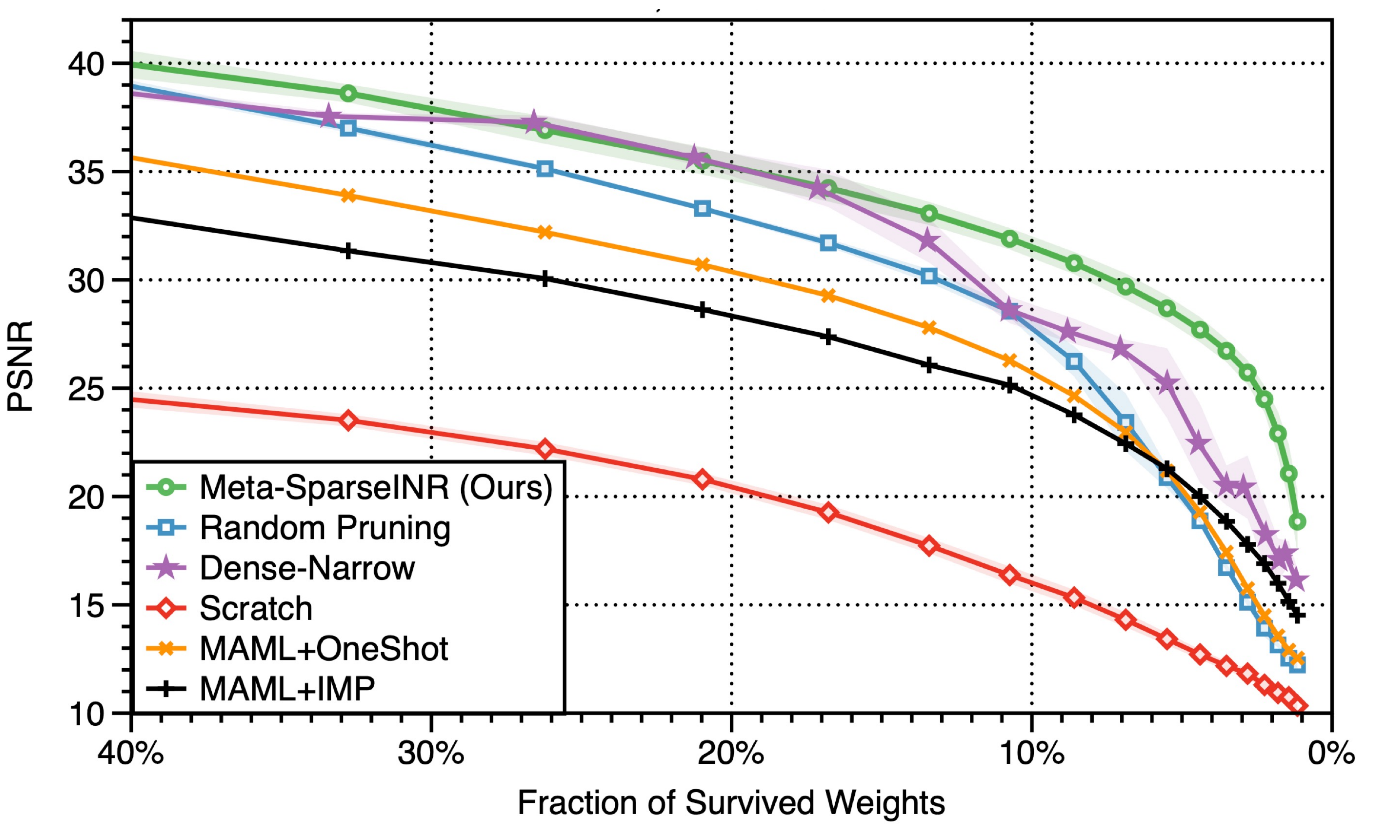


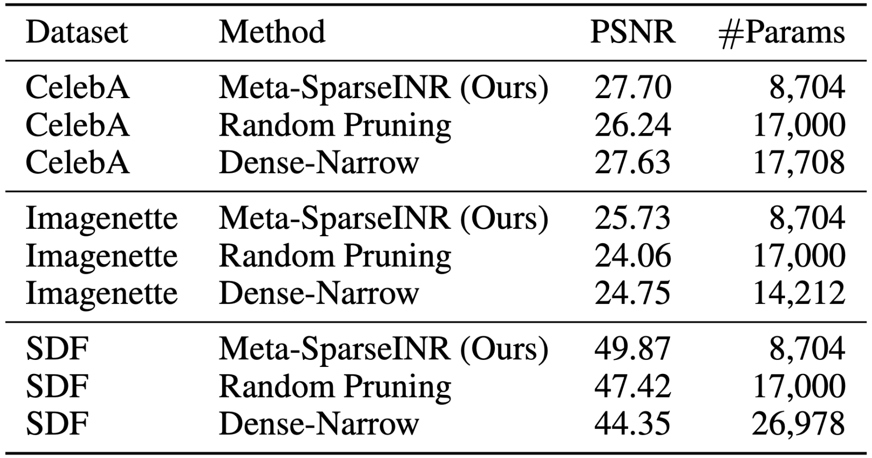
Idea: Meta-learned weights can be directly used as a pruning saliency score:

1. Meta-train a INR on a set of signals
2. Prune some connections based on meta-learned weights
3. Repeat

Experiments: Performance after 100 step training

SIREN on CelebA, when we can use 100 gradient steps for fitting each signal.



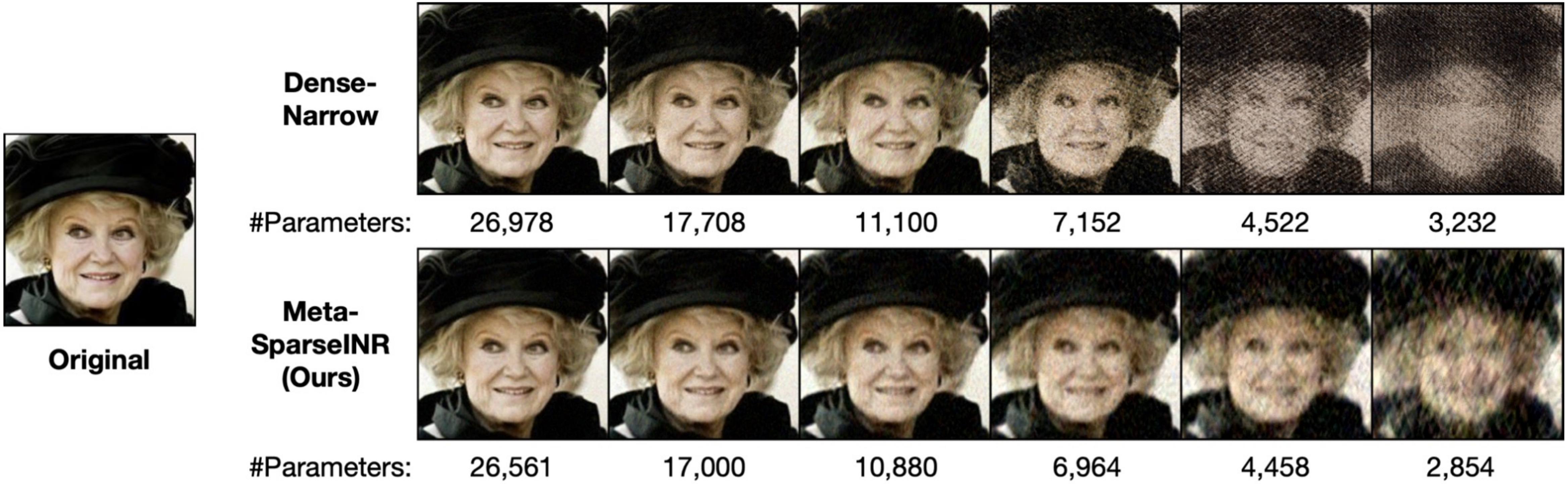


Dataset	Method	PSNR	#Params
CelebA	Meta-SparseINR (Ours)	27.70	8,704
CelebA	Random Pruning	26.24	17,000
CelebA	Dense-Narrow	27.63	17,708
Imagenette	Meta-SparseINR (Ours)	25.73	8,704
Imagenette	Random Pruning	24.06	17,000
Imagenette	Dense-Narrow	24.75	14,212
SDF	Meta-SparseINR (Ours)	49.87	8,704
SDF	Random Pruning	47.42	17,000
SDF	Dense-Narrow	44.35	26,978

Random Pruning: Same as Meta-SparseINR, but use the random pruning
Dense-Narrow: Meta-learn a dense neural representation that has a narrower width than the original INR

Experiments: Qualitative Comparisons

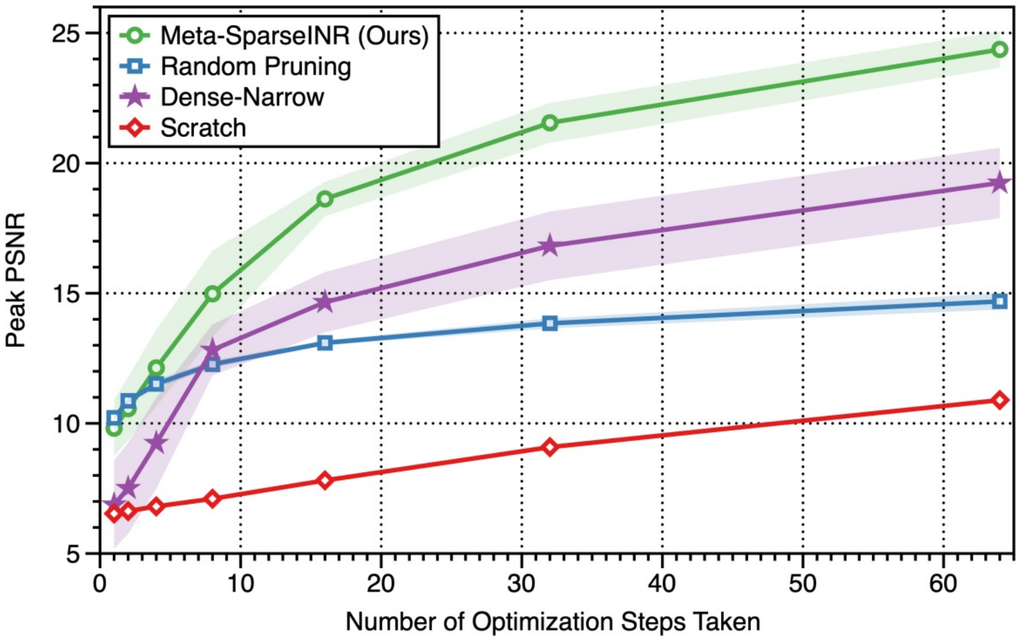
Interestingly, sparse INRs tend to give more “structured” outputs.


Original
Dense-Narrow
#Parameters: 26,978 17,708 11,100 7,152 4,522 3,232
Meta-SparseINR (Ours)
#Parameters: 26,561 17,000 10,880 6,964 4,458 2,854

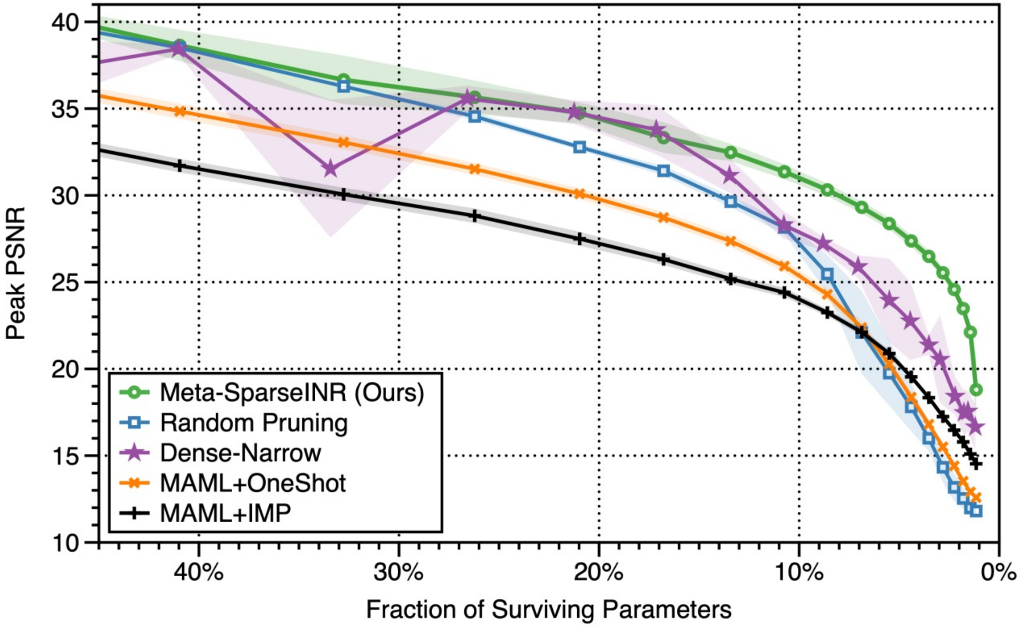
Experiments: Adaptation Efficiency and Cross-domain

(a) Meta-SparseINR learns the signal **much faster** compared to other baselines

(b) Moreover, our method even shows effectiveness on **cross-domain setup**



(a) PSNR vs. number of optimization steps



(b) Imagenette → CelebA

Summary of Meta-SparseINR

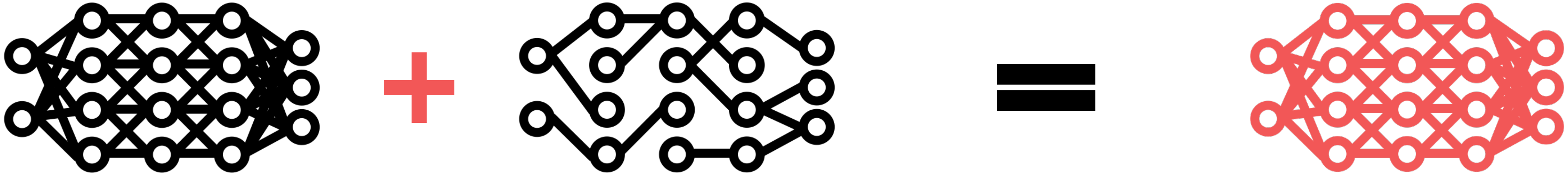
We develop a scalable method to learn
sparse neural representations for a **large set of signals**

We combine **meta-learning** and **network pruning** to train a sparse initial model

Recent Trends in Meta-Learning (Storage-Efficient) INRs

There exist various follow-up studies in this direction

- Common idea: **Only adapt few parameters** for the meta-learning



Initialization

Few-parameter update
(only save this part in the storage)

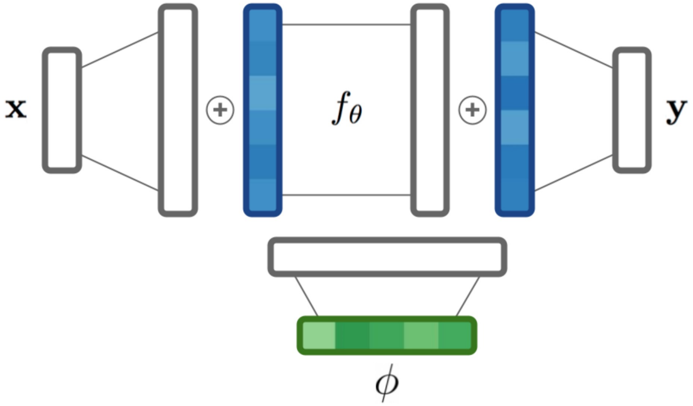
Target signal INR

Recent Trends in Meta-Learning (Storage-Efficient) INRs

There exist various follow-up studies in this direction

- Common idea: **Only adapt few parameters** for the meta-learning

Use the **shifting modulation** for the adaptation [1,2]



i^{th} layer of INR

$$\sin\left(\omega_0\left(\boxed{\mathbf{W}^{(i)}}\mathbf{x}+\boxed{\mathbf{b}^{(i)}}+\boxed{\mathbf{s}^{(i)}}\right)\right)$$

Globally shared parameters

Task-specific parameter
(shifting modulation)

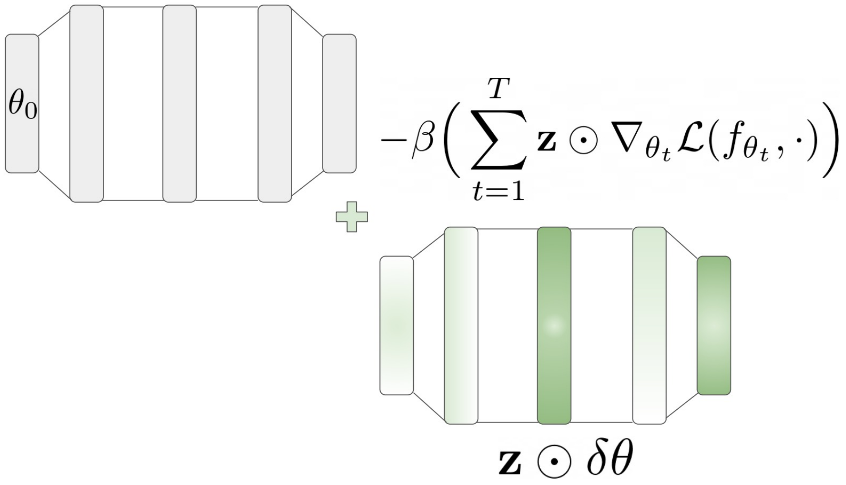
[1] Dupont et al. From data to functa: Your data point is a function and you should treat it like one. ICML 2022
[2] Dupont et al. COIN++: Neural Compression Across Modalities. arXiv 2022
[3] Schwarz et al. Meta-Learning Sparse Compression Networks, TMLR 2022

Recent Trends in Meta-Learning (Storage-Efficient) INRs

There exist various follow-up studies in this direction

- Common idea: **Only adapt few parameters** for the meta-learning

Use a **sparse gradient update** through ℓ_0 regularization [3]



θ_0
$-\beta\left(\sum_{t=1}^{T} \mathbf{z} \odot \nabla_{\theta_t} \mathcal{L}(f_{\theta_t}, \cdot)\right)$
$\mathbf{z} \odot \delta\theta$

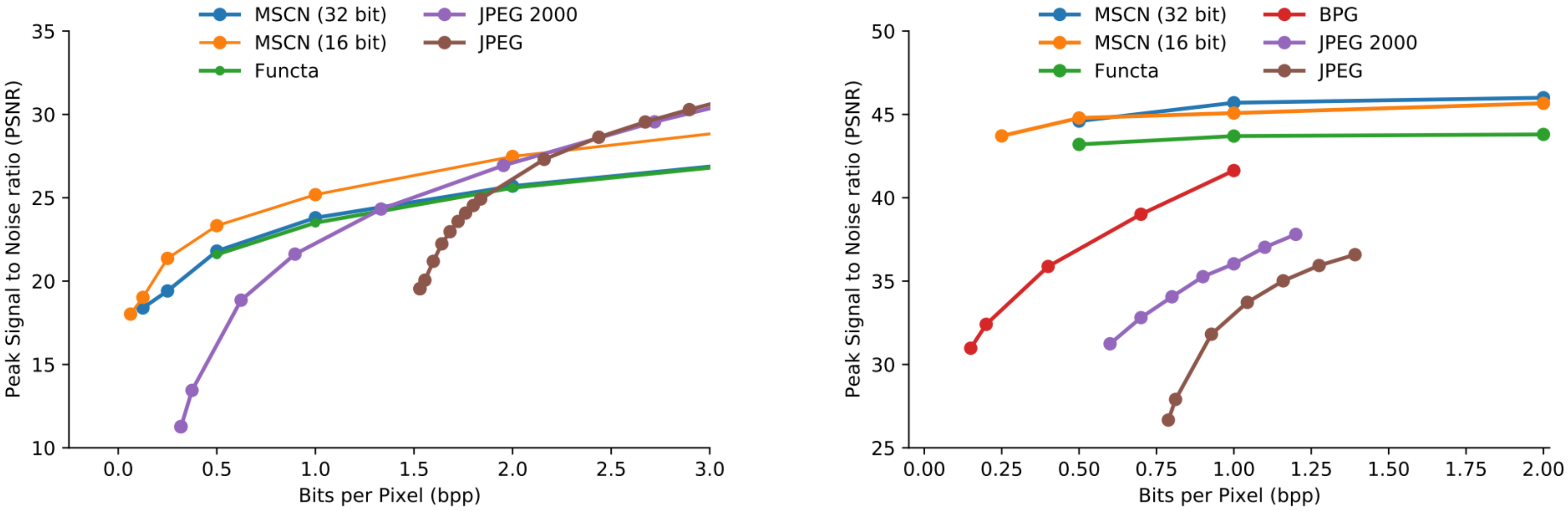
z: zero mask (learn from ℓ_0 regularization)

[1] Dupont et al. From data to functa: Your data point is a function and you should treat it like one. ICML 2022
[2] Dupont et al. COIN++: Neural Compression Across Modalities. arXiv 2022
[3] Schwarz et al. Meta-Learning Sparse Compression Networks, TMLR 2022

Recent Trends in Meta-Learning (Storage-Efficient) INRs

This research direction also can be extended to **data compression**!

- These approaches even show comparable performance with existing compression techniques, e.g., JPEG



(a) CelebA

(b) ERA 5

[1] Dupont et al. From data to functa: Your data point is a function and you should treat it like one. ICML 2022
[2] Dupont et al. COIN++: Neural Compression Across Modalities. arXiv 2022
[3] Schwarz et al. Meta-Learning Sparse Compression Networks, TMLR 2022

Summary

INR is an emerging paradigm for representing the data (or signals)

How to learn INRs for **a large set of signals** in an efficient manner?

Combine **meta-learning** and **parameter-efficient learning** schemes !

Thank you for your attention ☺