University of Southampton

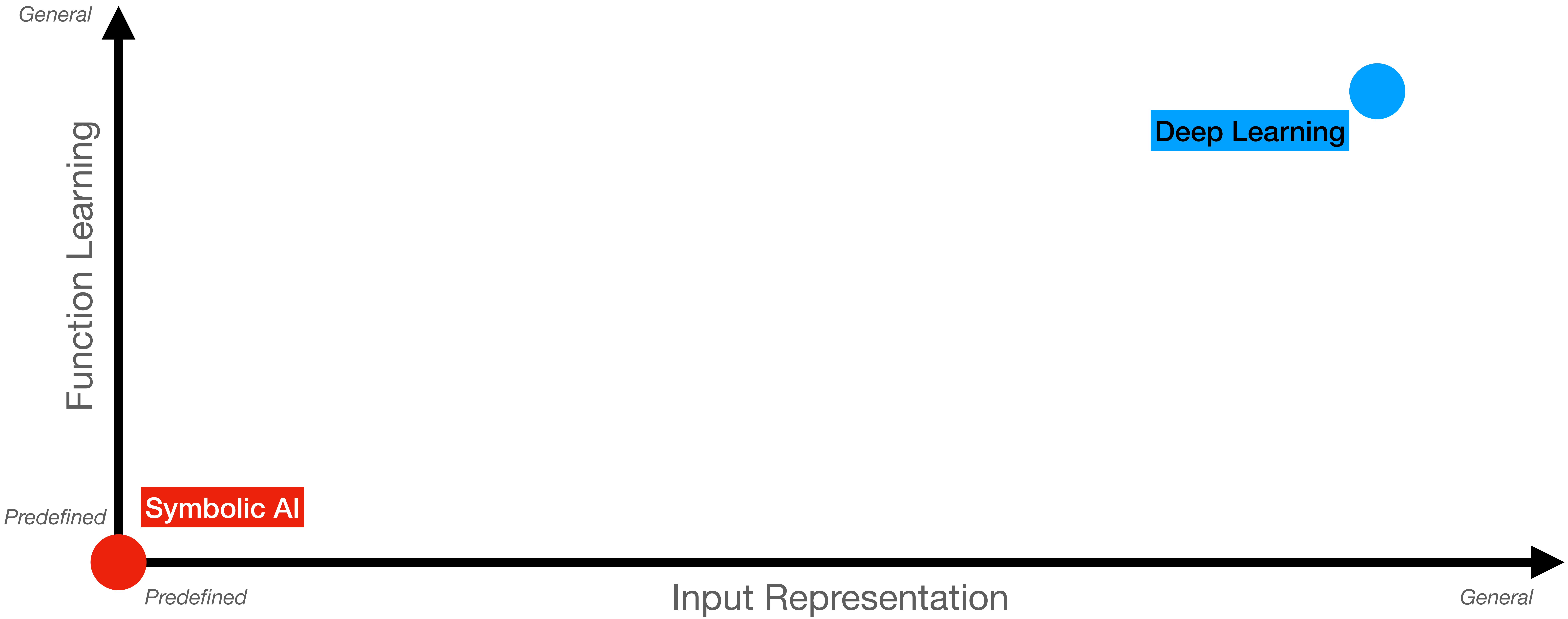
Incorporating constraints in learning machines through model architecture

Jonathon Hare, 27 October 2023

This presentation wouldn't be possible without:

(And Eduardo Manino for taking the photo!)

The AI Landscape



Credit to Antonia Creswell (https://neurips.cc/virtual/2021/22954) for this image

The AI Landscape

University of Southampton

General

Function Learning

Predefined

Symbolic AI Input: highly structured Function: highly structured	
Interpretability	🧠🧠🧠🧠🧠
Data Efficiency	🧠🧠🧠🧠🧠
Generalisation	🧠🧠🧠🧠
Universality	🧠
Extrapolation	🧠🧠🧠🧠🧠

Deep learning Input: (encoded) raw inputs* Function: weakly structured	
Interpretability	🧠
Data Efficiency	🧠
Generalisation	🧠🧠🧠
Universality	🧠🧠🧠🧠🧠
Extrapolation	🧠

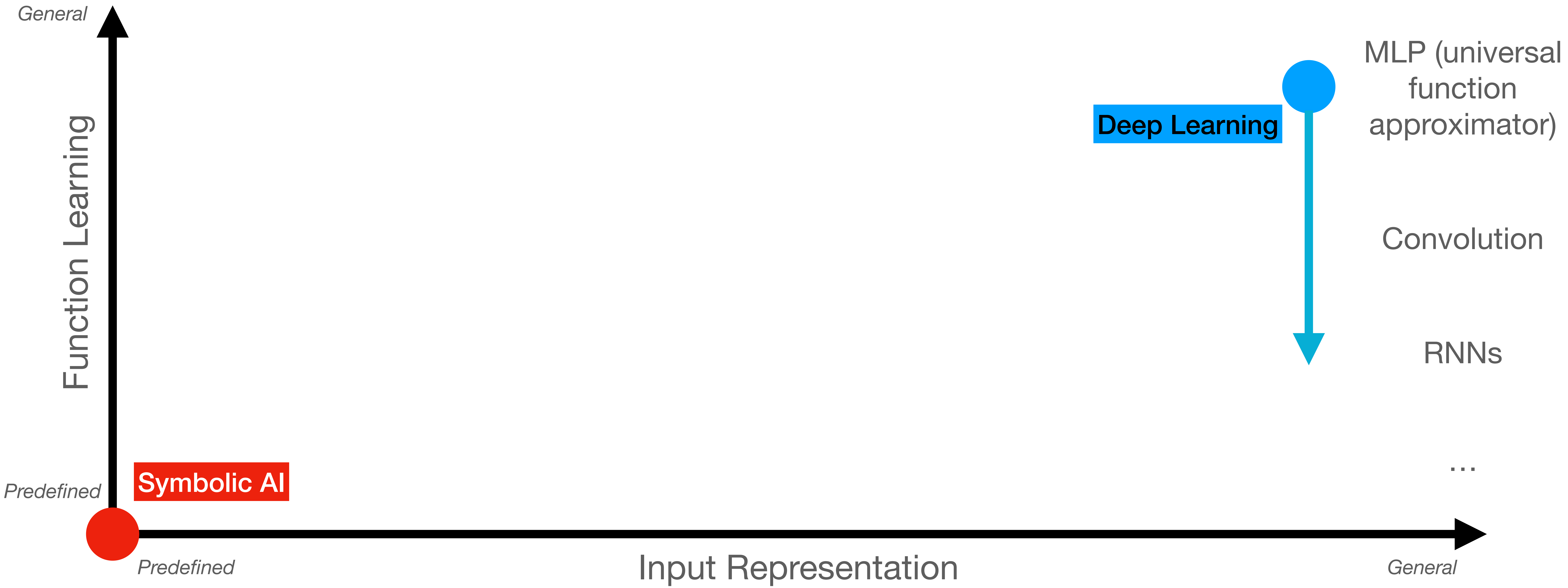
Predefined

Input Representation

General

Credit to Antonia Creswell (https://neurips.cc/virtual/2021/22954) for this image

(Deep) Learning Machines



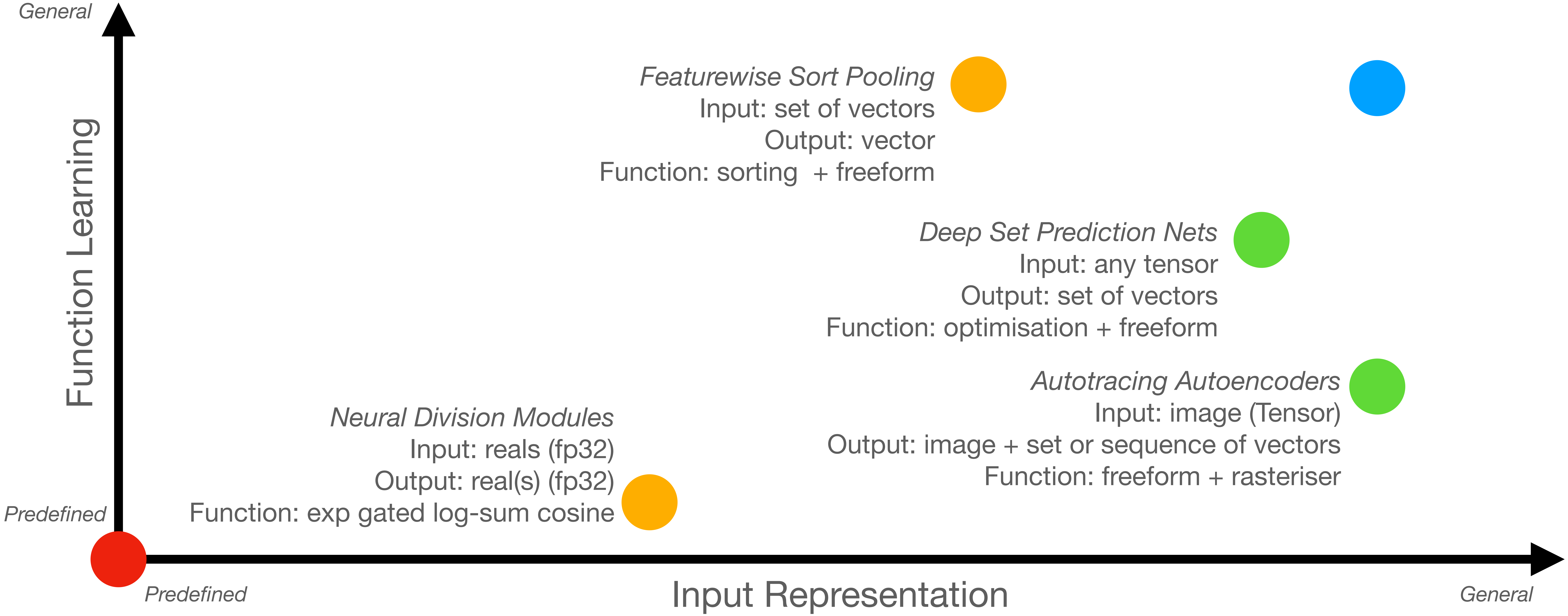
(Deep) Learning Machines

(Deep) Learning Machines

(Deep) Learning Machines

Thinking beyond matrix multiplication

Learning machines with architectural constraints



Autotracing Autoencoders

Daniela Mihai and Jonathon Hare (2021) Learning to Draw: Emergent Communication through Sketching. In Advances in Neural Information Processing Systems 34. vol. 34, Neural Information Processing Systems.

Daniela Mihai and Jonathon Hare (2021) Differentiable Drawing and Sketching. arxiv:2103.16194

Daniela Mihai and Jonathon Hare (2021) Perceptions. The AI Art Gallery: NeurIPS Workshop on Machine Learning for Creativity and Design 2021.

Daniela Mihai and Jonathon Hare (2021) Physically Embodied Deep Image Optimisation. 5th Workshop on Machine Learning for Creativity and Design (ML4CD 2021) of the Neural Information Processing Systems (NeurIPS), Virtual. 4 pp .

Daniela Mihai and Jonathon Hare (2021) Shared visual representations of drawing for communication: how do different biases affect human interpretability and intent? In Shared Visual Representations in Human and Machine Intelligence: 2021 NeurIPS Workshop. 10 pp .

Structured Image Generation

Machine vs human

- **Problem**: state of the art image generators use very unnatural processes

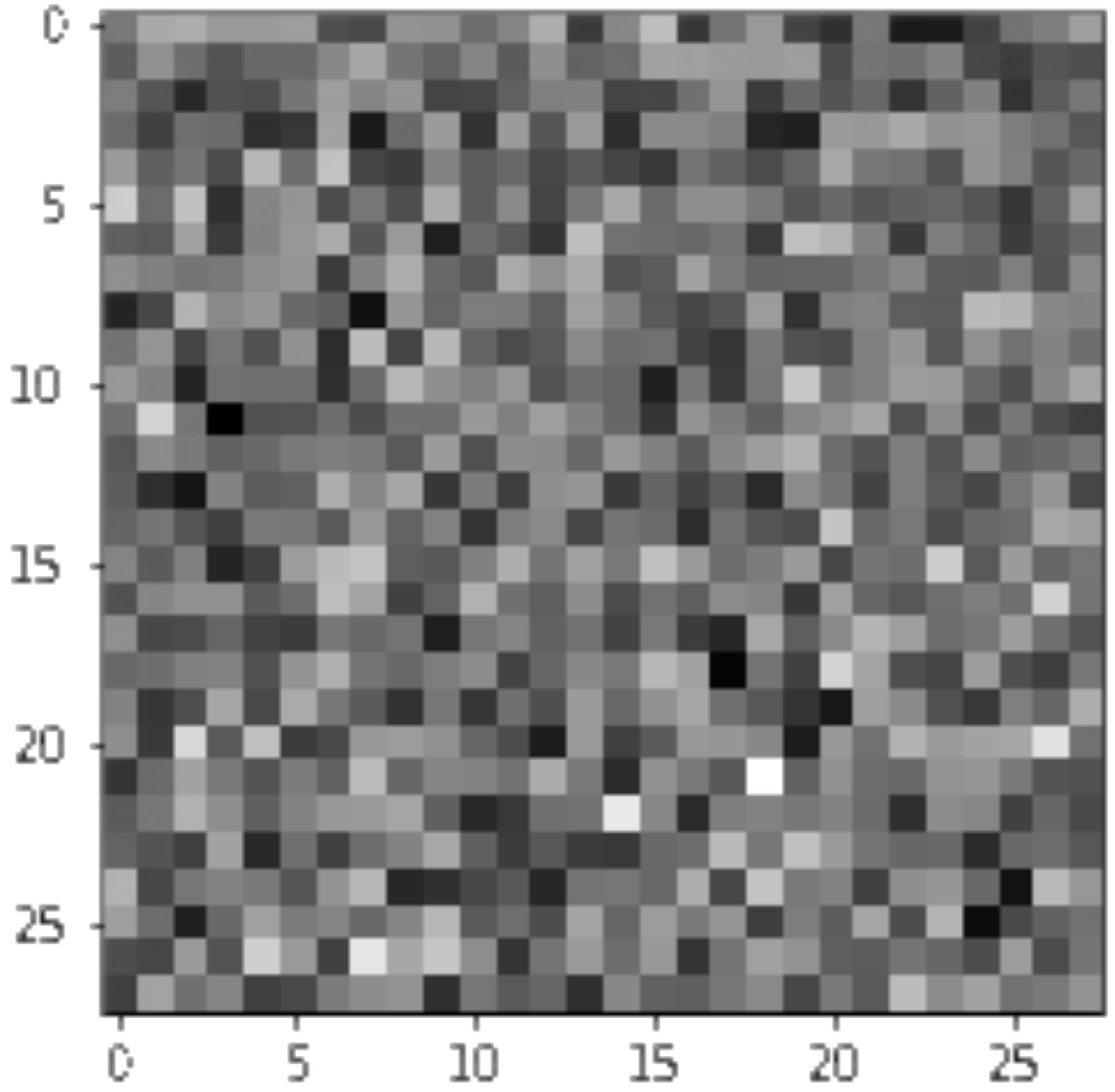


Image generation in deep learning involves functions that operate on all pixels simultaneously

https://huggingface.co/blog/annotated-diffusion

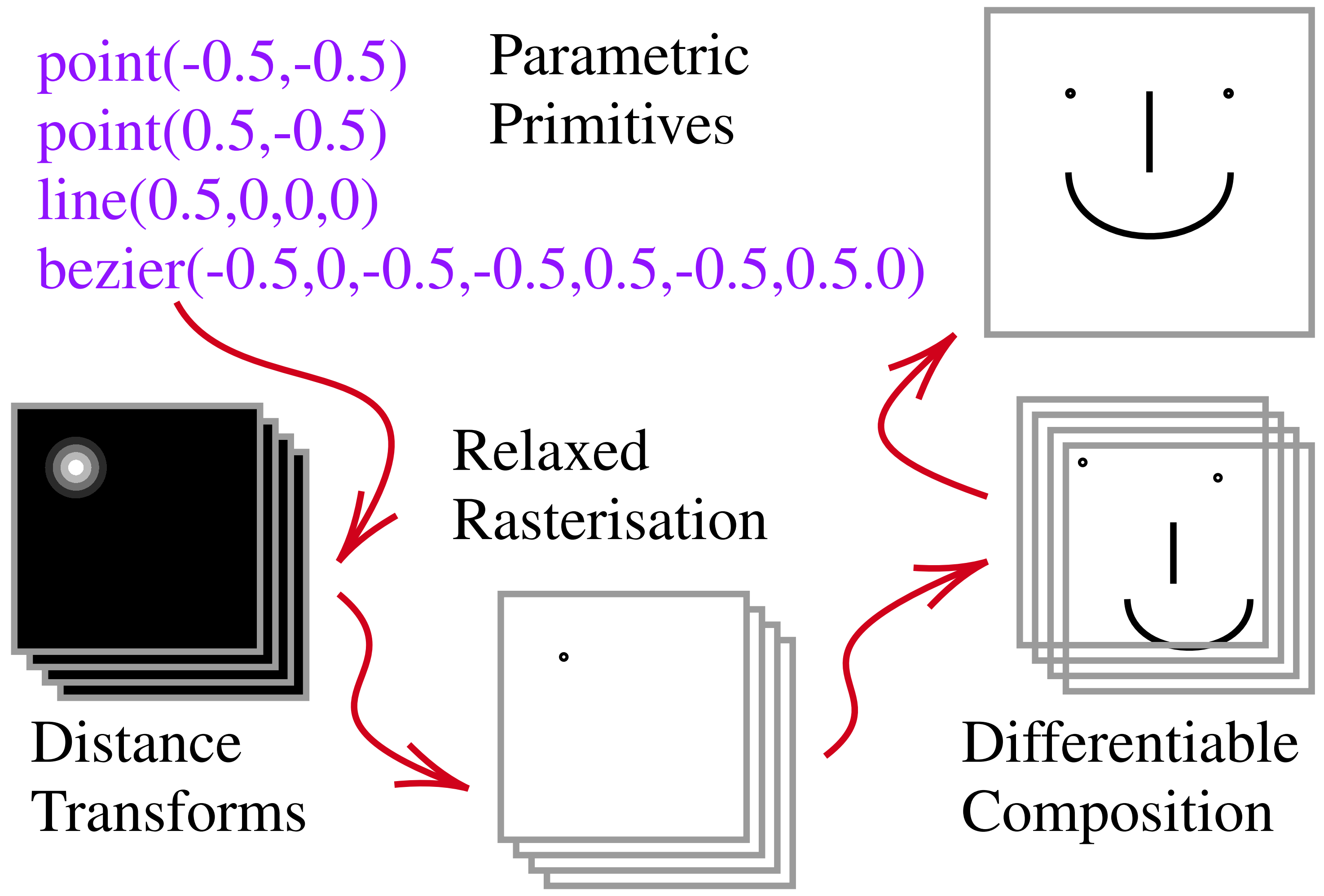
Humans generate images in a very different way - using tools to draw *strokes*

https://giphy.com/gifs/sketch-sketching-desenhar-4fxoQZNYnA8hsZzRNa

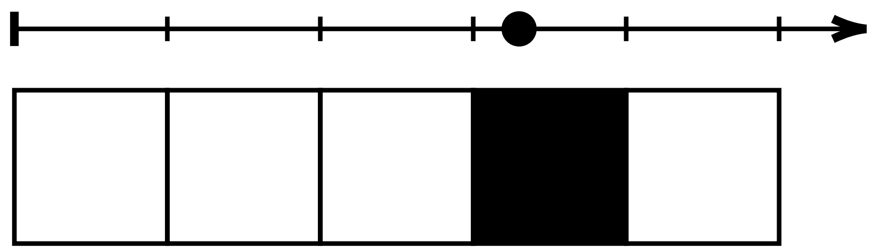
Structured line drawing

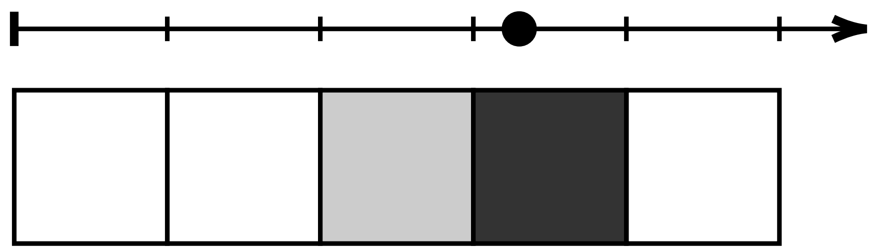
Differentiable rasterisation

- **Solution**: design an architecture that is constrained to draw a stroke and another that can compose strokes sequentially
  - Function needs to be incorporated in larger (deep) learning system -> must be differentiable
    - Standard rasterisation techniques are not differentiable
  - We need a differentiable relaxation of rasterisation…



Rasterisation (1d shown):





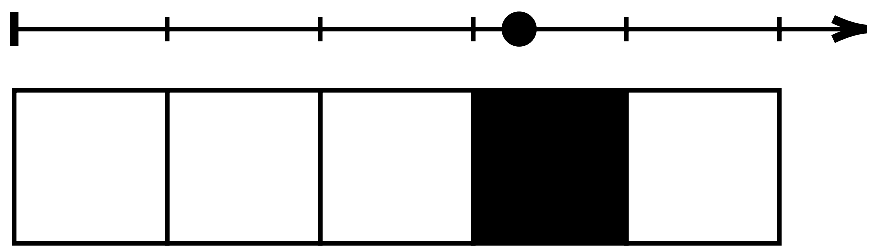
$$f(n;p) = \begin{cases} 1 & \text{if } \lfloor p \rfloor = n \\ 0 & \text{otherwise .} \end{cases}$$

$$f(n;p) = \begin{cases} 1.5 - p + \lfloor p - 0.5 \rfloor & \text{if } \lfloor p - 0.5 \rfloor = n \\ 0.5 + p - \lceil p - 0.5 \rceil & \text{if } \lceil p - 0.5 \rceil = n \\ 0 & \text{otherwise .} \end{cases}$$

Image Composition:

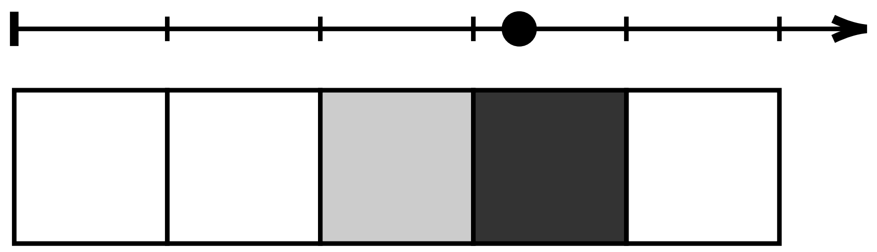
$$c(\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \ldots, \boldsymbol{I}^{(n)}) = \boldsymbol{I}^{(1)} \vee \boldsymbol{I}^{(2)} \vee \ldots \vee \boldsymbol{I}^{(n)}$$

Rasterisation (1d shown):



$$f(n;p) = \begin{cases} 1 & \text{if } \lfloor p \rfloor = n \\ 0 & \text{otherwise .} \end{cases}$$

No good gradient



$$f(n;p) = \begin{cases} 1.5 - p + \lfloor p - 0.5 \rfloor & \text{if } \lfloor p - 0.5 \rfloor = n \\ 0.5 + p - \lceil p - 0.5 \rceil & \text{if } \lceil p - 0.5 \rceil = n \\ 0 & \text{otherwise .} \end{cases}$$

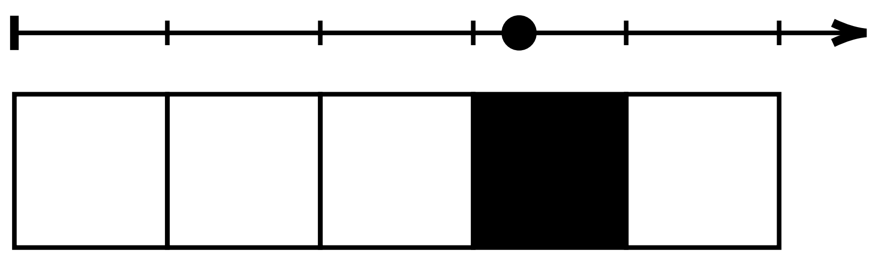
No good gradient

Image Composition:

$$c(\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \ldots, \boldsymbol{I}^{(n)}) = \boldsymbol{I}^{(1)} \vee \boldsymbol{I}^{(2)} \vee \ldots \vee \boldsymbol{I}^{(n)}$$

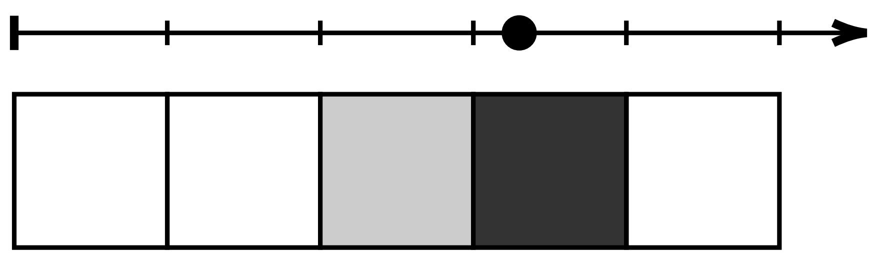
No good gradient

Rasterisation (1d shown):



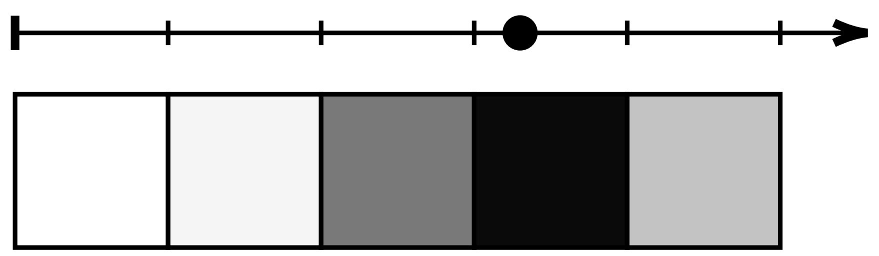
$$f(n;p) = \begin{cases} 1 & \text{if } \lfloor p \rfloor = n \\ 0 & \text{otherwise .} \end{cases}$$

No good gradient



$$f(n;p) = \begin{cases} 1.5 - p + \lfloor p - 0.5 \rfloor & \text{if } \lfloor p - 0.5 \rfloor = n \\ 0.5 + p - \lceil p - 0.5 \rceil & \text{if } \lceil p - 0.5 \rceil = n \\ 0 & \text{otherwise .} \end{cases}$$

No good gradient



$$f(n;p) = \exp(-d^2(n, p - 0.5)/\sigma^2)$$

Good gradient

Image Composition:

$$c(\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \ldots, \boldsymbol{I}^{(n)}) = \boldsymbol{I}^{(1)} \vee \boldsymbol{I}^{(2)} \vee \ldots \vee \boldsymbol{I}^{(n)}$$

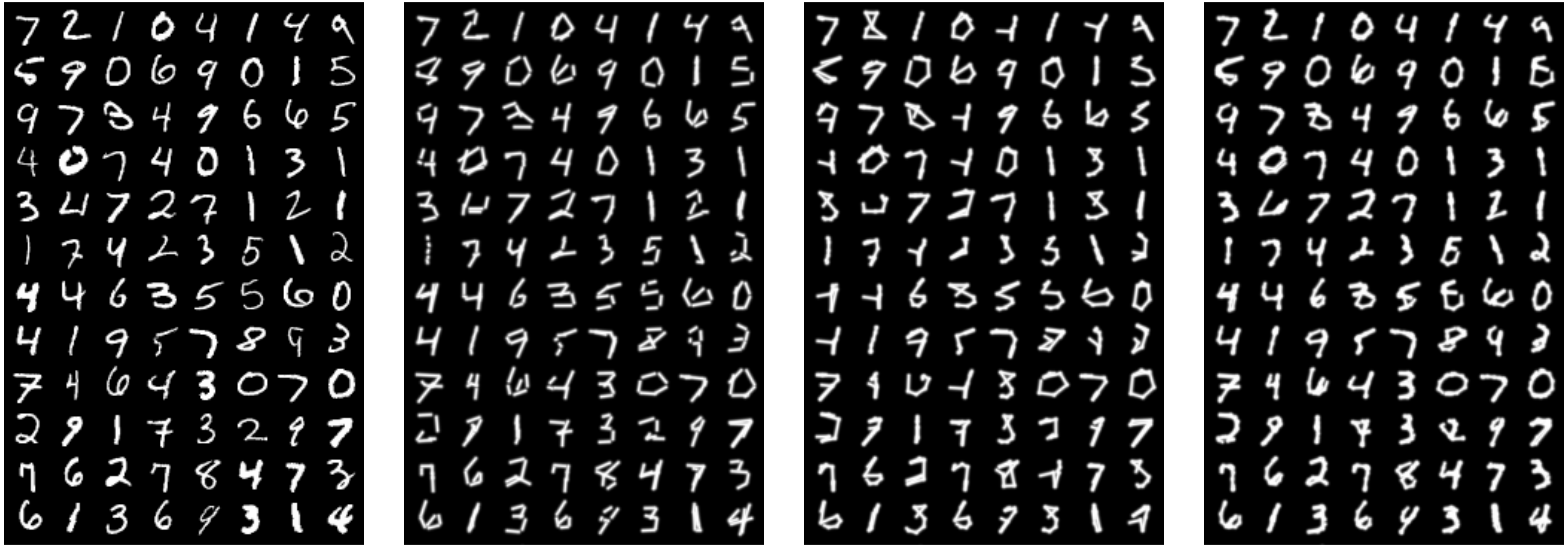
No good gradient

$$c_{\text{softor}}(\boldsymbol{I}^{(1)}, \boldsymbol{I}^{(2)}, \ldots, \boldsymbol{I}^{(n)}) = \mathbf{1} - \prod_{i=1}^{n} (\mathbf{1} - \boldsymbol{I}^{(i)})$$

Good gradient

Autotracing Autoencoders

Learning to convert drawings to vectors



(a) Test Samples (b) Lines(L=5) (c) PolyLine(P=8) (d) PolyLine(P=16)

Decoder	#P	#S	#L	MSE	Acc.
Line	10	1	5	0.0195	94.06%
PolyLine	16	15	1	0.0225	93.27%
PolyConnect	16	-	-	0.0118	96.47%
CRS	16	14	1	0.0208	94.63%
Bézier	20	1	5	0.0136	96.34%
BézierConnect	16	-	-	0.0116	96.43%

(a) MNIST Test Dataset (baseline unencoded acc. 98.60%).

Model	Steps	#P	#S	#L	Acc.
StrokeNet [39]	3 (SN)	16	14	1	95.25%
StrokeNet [39]	1	16	14	1	97.75%
Ours, CRS	1	16	14	1	97.12%
Ours, Bézier	3 (GRU)	4	1	1	96.97%
Ours, Bézier	1	7	2	2	98.28%
Ours, Bézier	1	43	14	1	97.94%

(b) Scaled MNIST Dataset (baseline unencoded acc. 98.58%).

University of Southampton

Generating Sets

Yan Zhang, Jonathon Hare and Adam Prugel-Bennett (2019) Deep Set Prediction Networks. In Advances in Neural Information Processing Systems 32. vol. 32, Neural Information Processing Systems.

Yan Zhang and Jonathon Hare and Adam Prügel-Bennett (2020) FSPool: Learning Set Representations with Featurewise Sort Pooling. International Conference on Learning Representations.

Yan Zhang, Jonathon Hare and Adam Prugel-Bennett (2019) FSPool: Learning set representations with featurewise sort pooling. Sets & Partitions: NeurIPS 2019 Workshop

Yan Zhang, Jonathon Hare and Adam Prugel-Bennett (2019) Deep Set Prediction Networks. Sets & Partitions: NeurIPS 2019 Workshop

Predicting sets

Learning unordered things with an ordered function is hard

- **Problem**: turn a vector (or more generally tensor) into a set of vectors
  - **Applications**: predicting objects in images, molecule generation, …
  - But, MLPs have ordered outputs and sets are by definition unordered

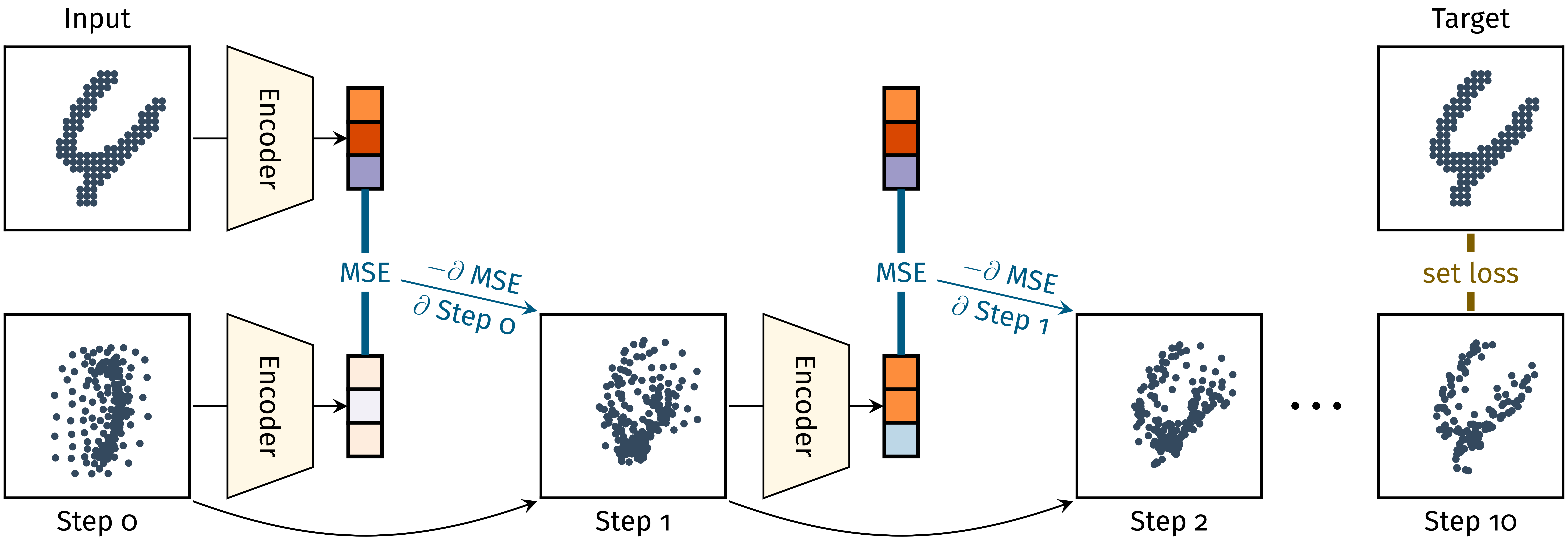
Reversing an invariant encoder

Deep Set Prediction Networks

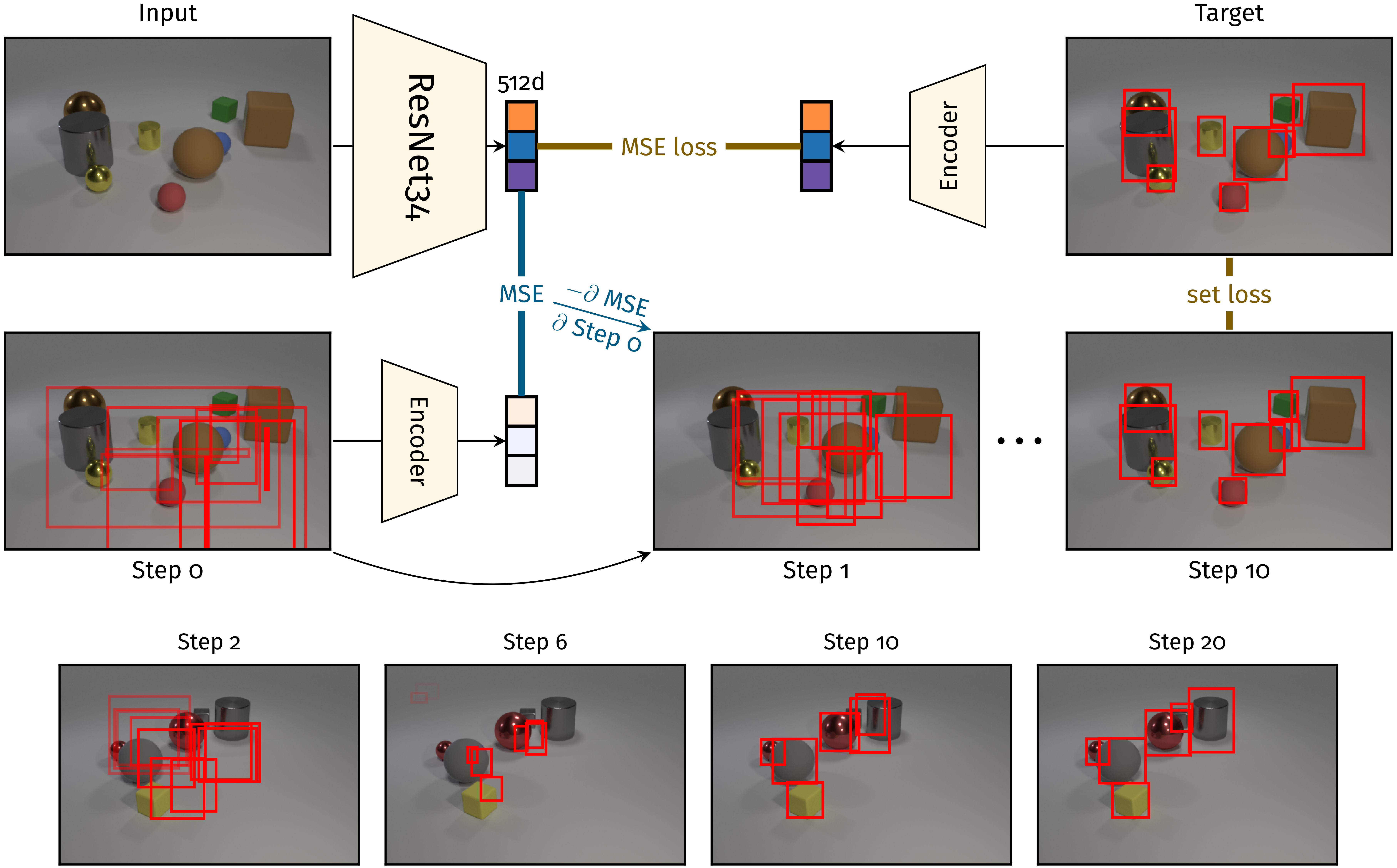
University of Southampton

- **Solution**: need to define a function (or procedure) that is *unordered*
  - **Observation**: gradient of a permutation-invariant encoder (set to vector) with respect to the input are permutation equivariant
    - i.e. gradients $\frac{\delta \text{loss}}{\delta \text{set}}$ do not depend on order!
  - **Implication**: to decode a feature vector into a set, we can use gradient descent to find a set that encodes to that feature vector
    - We can define a procedure that iteratively follows gradients in the forward pass

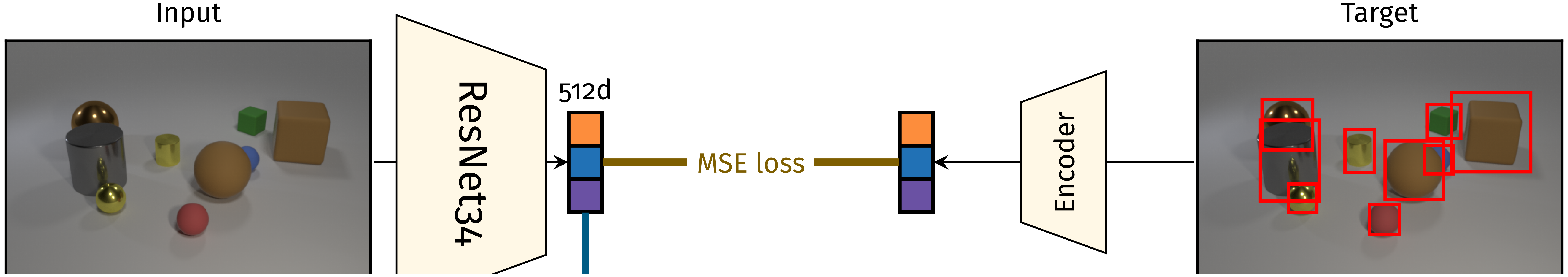
Autoencoding sets



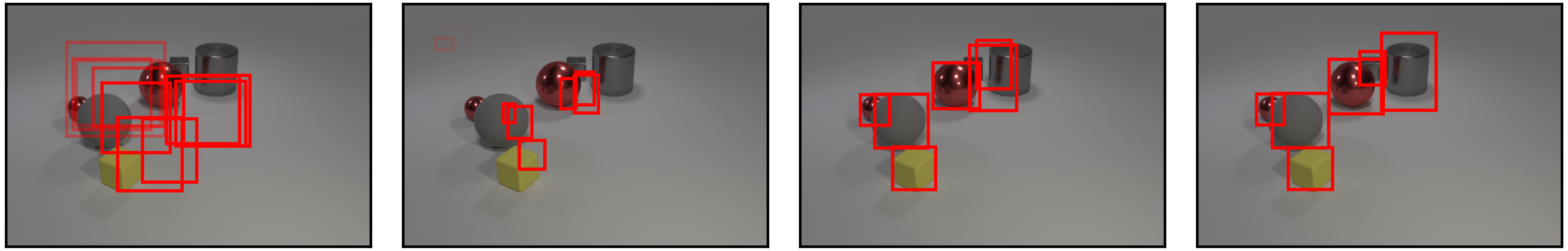
Object detection



Object detection

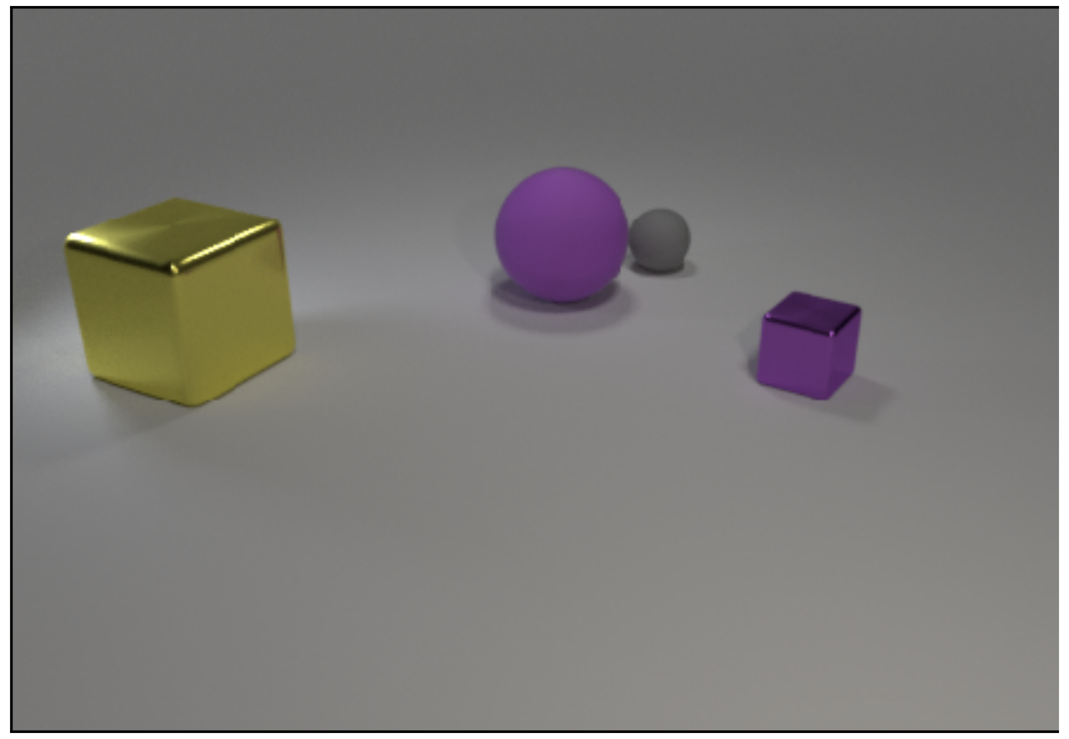


Bounding box prediction	AP_{50}	AP_{90}	AP_{95}	AP_{98}	AP_{99}
MLP baseline	$99.3_{\pm 0.2}$	$94.0_{\pm 1.9}$	$57.9_{\pm 7.9}$	$0.7_{\pm 0.2}$	$0.0_{\pm 0.0}$
RNN baseline	$99.4_{\pm 0.2}$	$94.9_{\pm 2.0}$	$65.0_{\pm 10.3}$	$2.4_{\pm 0.0}$	$0.0_{\pm 0.0}$
Ours (train 10 steps, eval 10 steps)	$98.8_{\pm 0.3}$	$94.3_{\pm 1.5}$	$85.7_{\pm 3.0}$	$\mathbf{34.5}_{\pm 5.7}$	$\mathbf{2.9}_{\pm 1.2}$
Ours (train 10 steps, eval 20 steps)	$\mathbf{99.8}_{\pm 0.0}$	$\mathbf{98.7}_{\pm 1.1}$	$\mathbf{86.2}_{\pm 7.2}$	$24.3_{\pm 8.0}$	$1.4_{\pm 0.9}$
Ours (train 10 steps, eval 30 steps)	$\mathbf{99.8}_{\pm 0.1}$	$96.7_{\pm 2.4}$	$75.5_{\pm 12.3}$	$17.4_{\pm 7.7}$	$0.9_{\pm 0.7}$



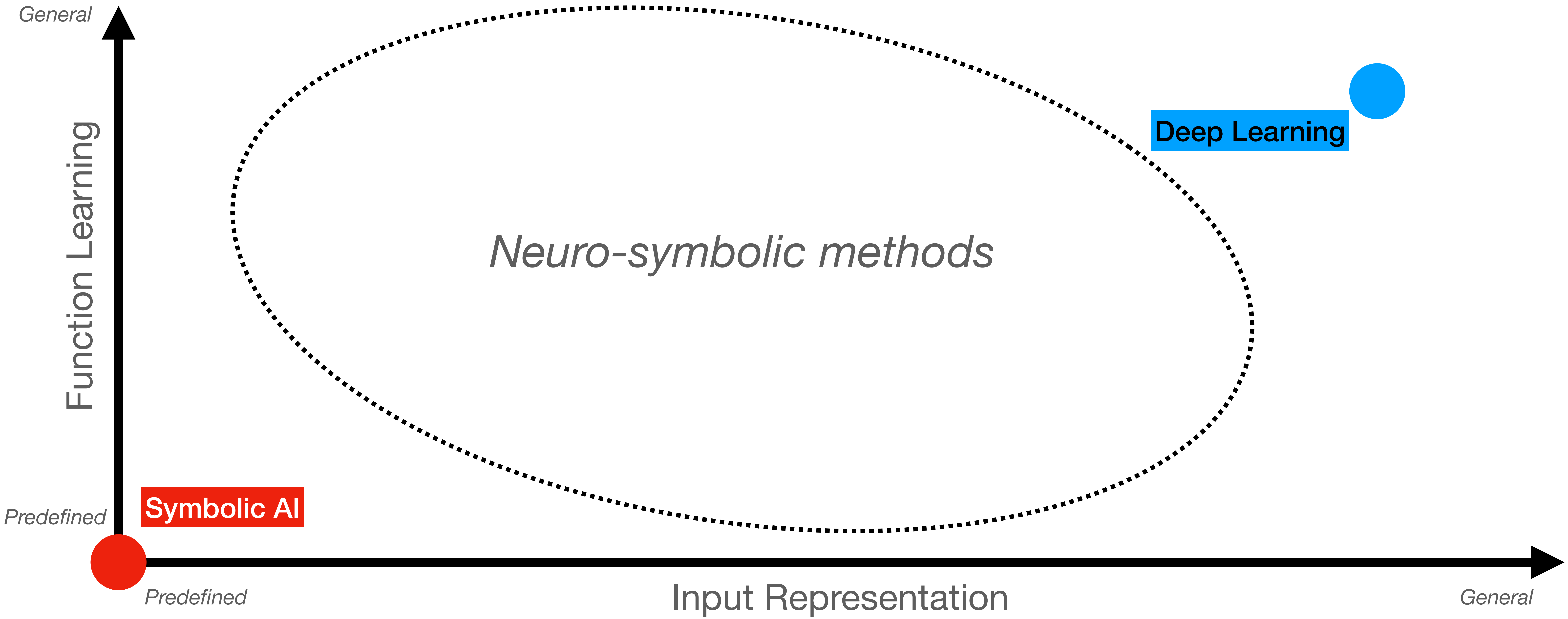
Object and attribute prediction

Object attribute prediction	AP_{∞}	AP_1	$AP_{0.5}$	$AP_{0.25}$	$AP_{0.125}$
MLP baseline	$3.6_{\pm 0.5}$	$1.5_{\pm 0.4}$	$0.8_{\pm 0.3}$	$0.2_{\pm 0.1}$	$0.0_{\pm 0.0}$
RNN baseline	$4.0_{\pm 1.9}$	$1.8_{\pm 1.2}$	$0.9_{\pm 0.5}$	$0.2_{\pm 0.1}$	$0.0_{\pm 0.0}$
Ours (train 10 steps, eval 10 steps)	$72.8_{\pm 2.3}$	$59.2_{\pm 2.8}$	$39.0_{\pm 4.4}$	$12.4_{\pm 2.5}$	$1.3_{\pm 0.4}$
Ours (train 10 steps, eval 20 steps)	$84.0_{\pm 4.5}$	$80.0_{\pm 4.9}$	$\mathbf{57.0}_{\pm 12.1}$	$\mathbf{16.6}_{\pm 9.0}$	$\mathbf{1.6}_{\pm 0.9}$
Ours (train 10 steps, eval 30 steps)	$\mathbf{85.2}_{\pm 4.8}$	$\mathbf{81.1}_{\pm 5.2}$	$47.4_{\pm 17.6}$	$10.8_{\pm 9.0}$	$0.6_{\pm 0.7}$

Input	Step 5	Step 10	Step 20	Target
	x, y, z = (-0.14, 1.16, 3.57) large purple rubber sphere	x, y, z = (-2.33, -2.41, 0.73) large yellow metal cube	x, y, z = (-2.33, -2.42, 0.78) large yellow metal cube	x, y, z = (-2.42, -2.40, 0.70) large yellow metal cube
	x, y, z = (0.01, 0.12, 3.42) large gray metal cube	x, y, z = (-1.20, 1.27, 0.67) large purple rubber sphere	x, y, z = (-1.21, 1.20, 0.65) large purple rubber sphere	x, y, z = (-1.18, 1.25, 0.70) large purple rubber sphere
	x, y, z = (0.67, 0.65, 3.38) small purple metal cube	x, y, z = (-0.96, 2.54, 0.36) small gray rubber sphere	x, y, z = (-0.96, 2.59, 0.36) small gray rubber sphere	x, y, z = (-1.02, 2.61, 0.35) small gray rubber sphere
	x, y, z = (0.67, 1.14, 2.96) small purple rubber sphere	x, y, z = (1.61, 1.57, 0.36) small yellow metal cube	x, y, z = (1.58, 1.62, 0.38) small purple metal cube	x, y, z = (1.74, 1.53, 0.35) small purple metal cube

Where next?

The AI Landscape



Credit to Antonia Creswell (https://neurips.cc/virtual/2021/22954) for this image

What do we want from learning AI?

- Good generalisation
    - ID / OOD
- Good robustness
    - Not falling for trivial adversarial examples
- Good explainability
    - Some understanding of why a machine is making a decision
    - Well calibrated confidences

What do we want from learning AI?

University of Southampton

- Good generalisation
  - ID / OOD
- Good robustness
  - Not falling for trivial adversarial examples
- Good explainability
  - Some understanding of why a machine is making a decision
  - Well calibrated confidences

How do we measure these?

Many (most?) current measures are flawed

What do we want from learning AI?

University of Southampton

- Good generalisation
  - ID / OOD
- Good robustness
  - Not falling for trivial adversarial examples
- Good explainability
  - Some understanding of why a machine is making a decision
  - Well calibrated confidences

The importance of feature combinations - (highly nonlinear) decision rules integrating information from different sources

Cue combination (cognitive sciences)
Disentanglement and compositionality [in a semantic sense]
"Decision Decompositionality"
"More distributed features"
Entanglement [in a geometric sense]
Function re-use

What do we want from learning AI?

- Good generalisation
  - ID / OOD
- Good robustness
  - Not falling for trivial adversarial examples
- Good explainability
  - Some understanding of why a machine is making a decision
  - Well calibrated confidences

My take:

We need to architecture functions to achieve these goals

Playing with losses can help, but will only get us so far - our models might learn, but maybe not in the intended direction

Any Questions?