Research Project Proposal:

Structured Learning

Alberto Archetti
alberto1.archetti@mail.polimi.it
Computer Science and Engineering

POLITECNICO
MILANO 1863

HONOURS PROGRAMME
HP-SR
in Information Technology

1. Deep Learning
2. Structured Learning
3. Autonomous Navigation

1. Deep Learning
2. Structured Learning
3. Autonomous Navigation

Neural Networks, Backpropagation

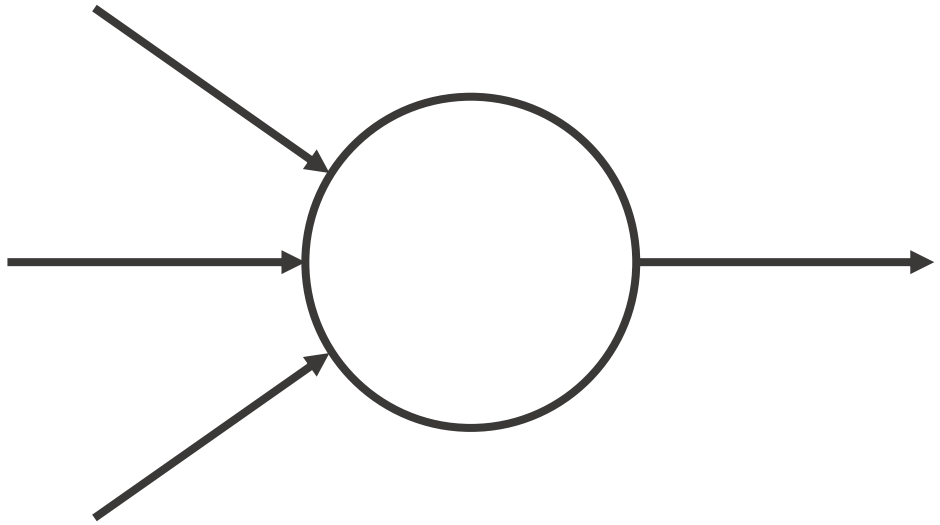
Deep Learning, drawbacks

Image classification

How can we teach a computer to recognize cats versus dogs?

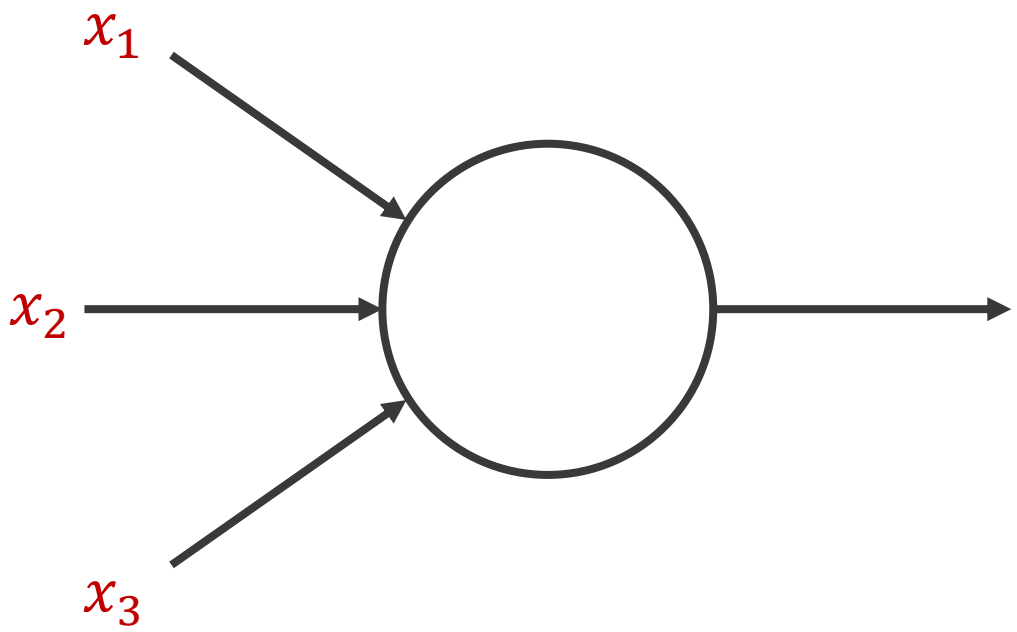
HP@CSE, Alberto Archetti

Building blocks



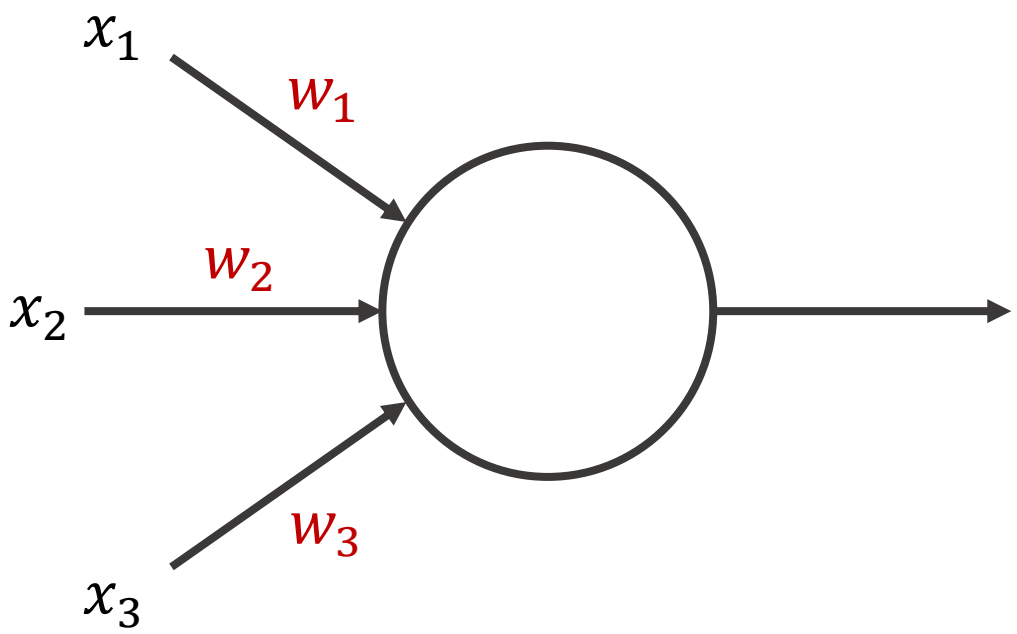
Neuron

Building blocks



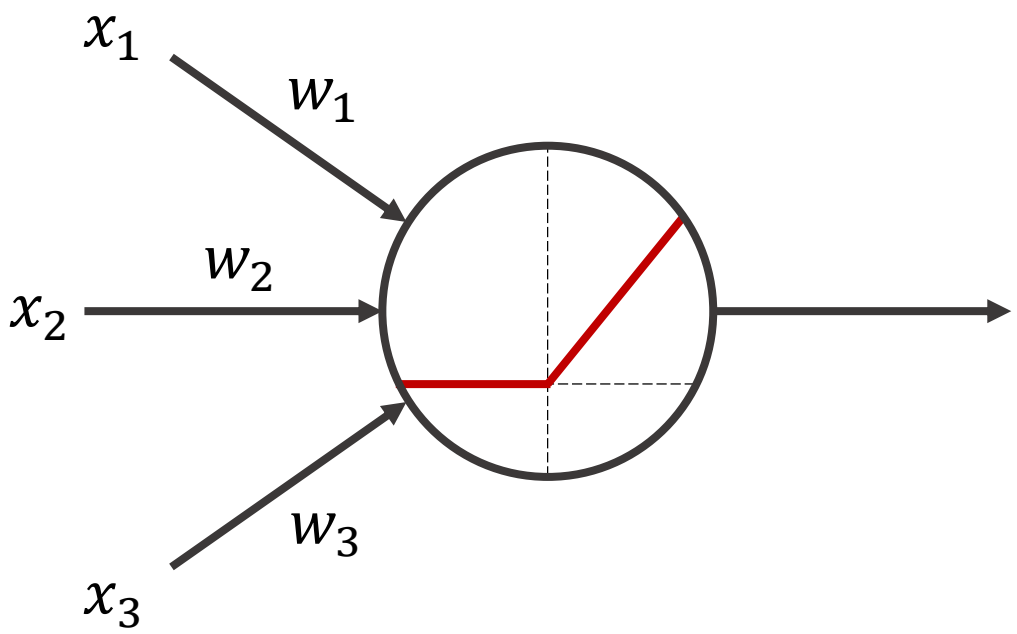
Inputs

Building blocks



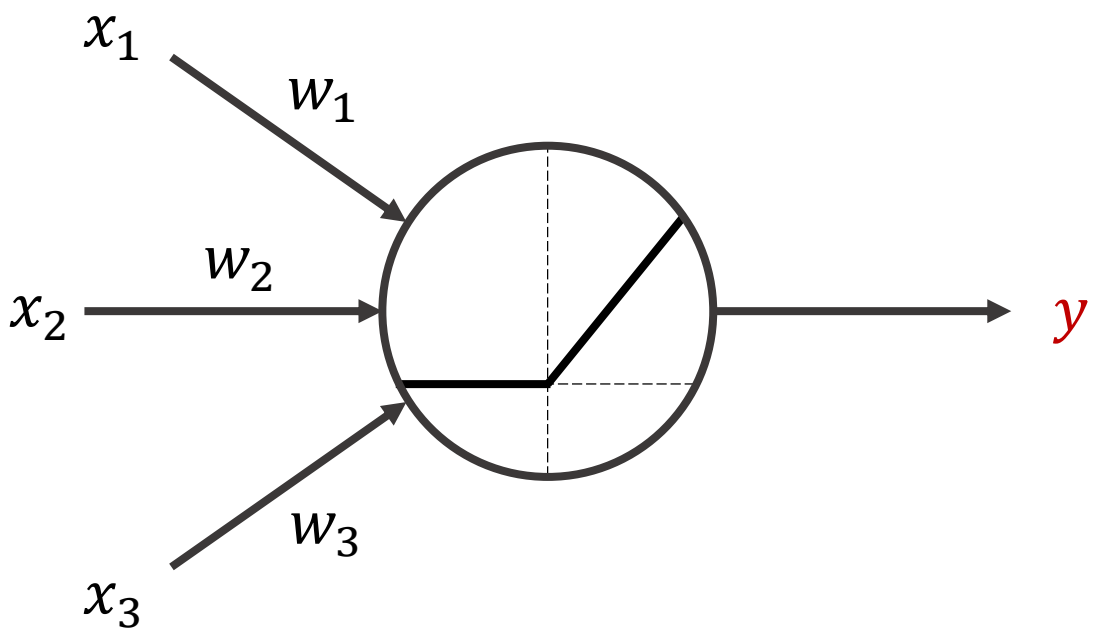
Weights

Building blocks

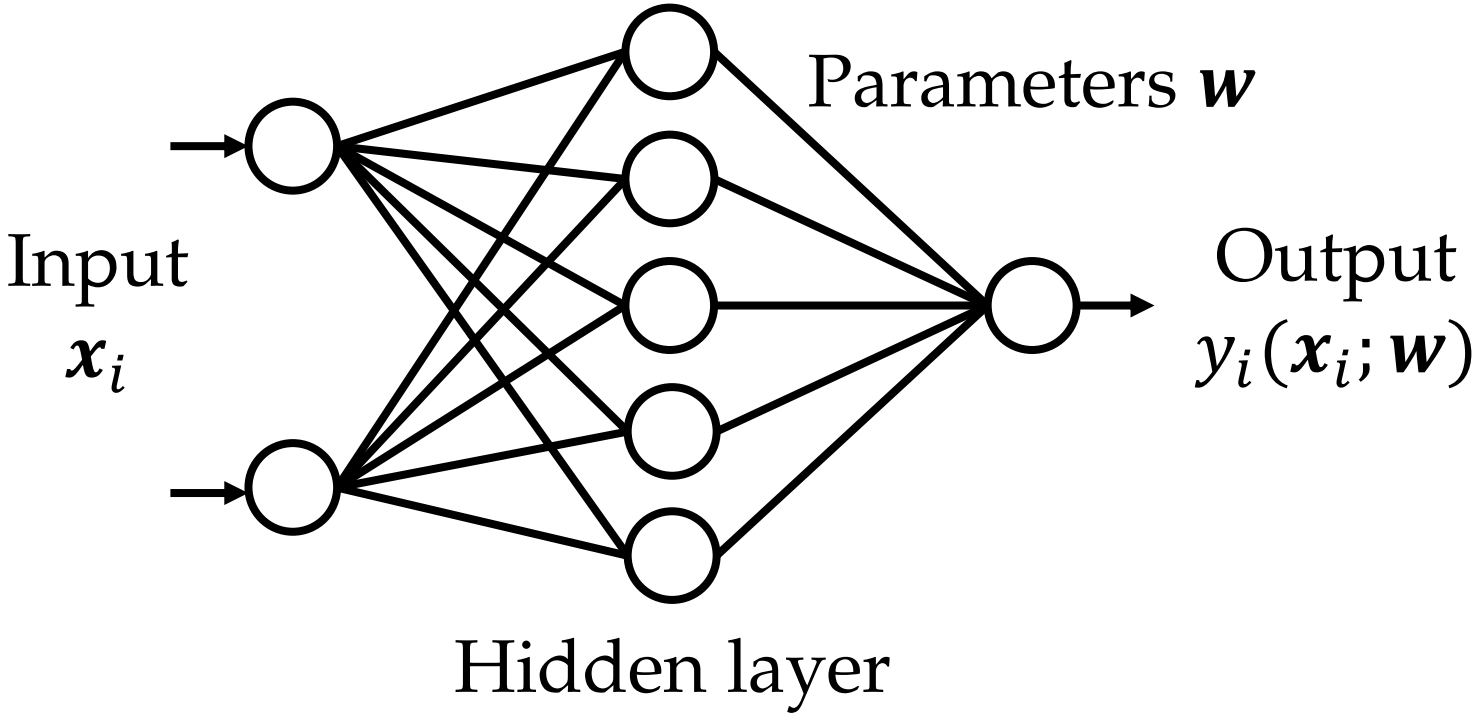


Activation function

Building blocks

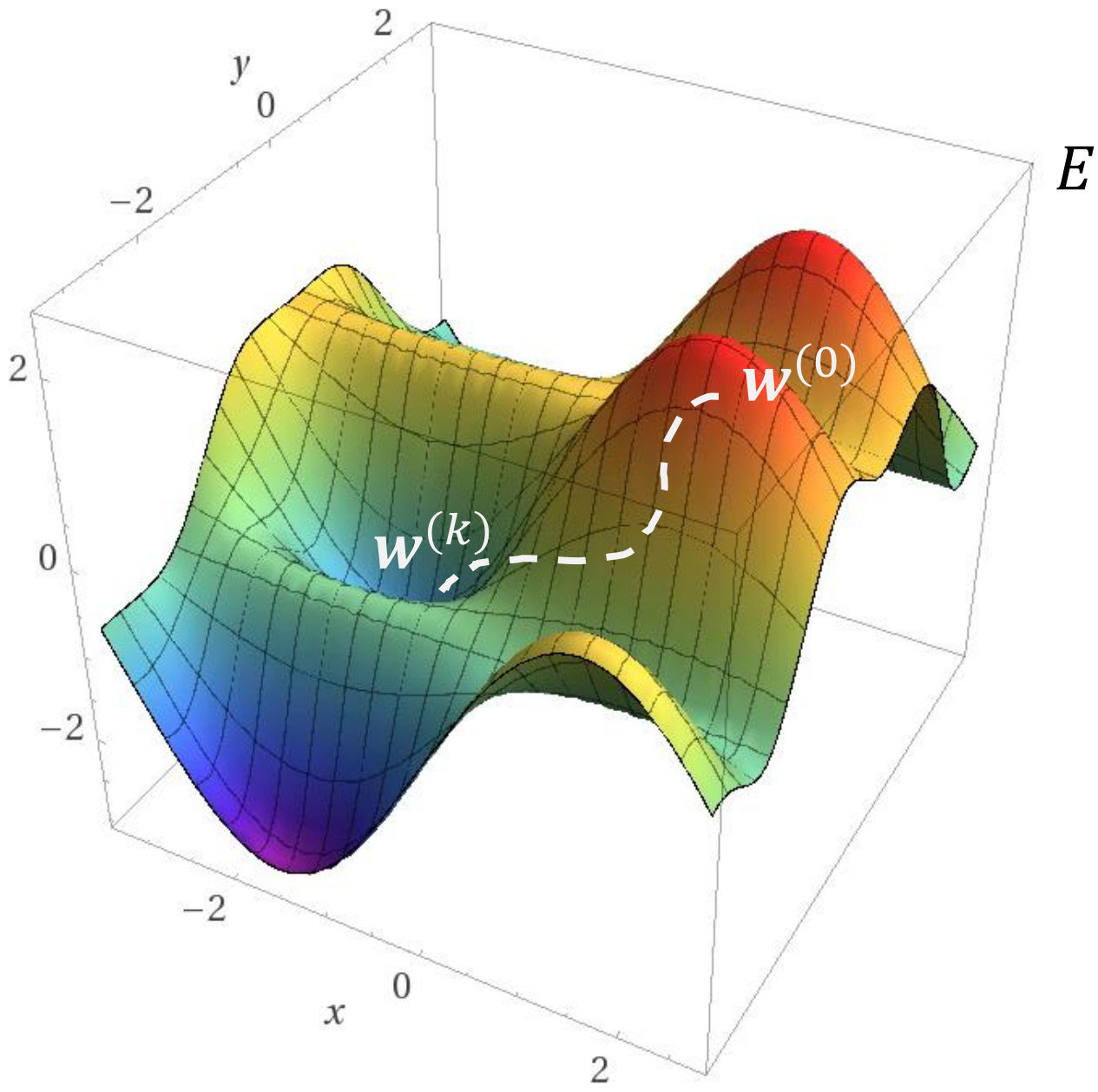


Neural Network

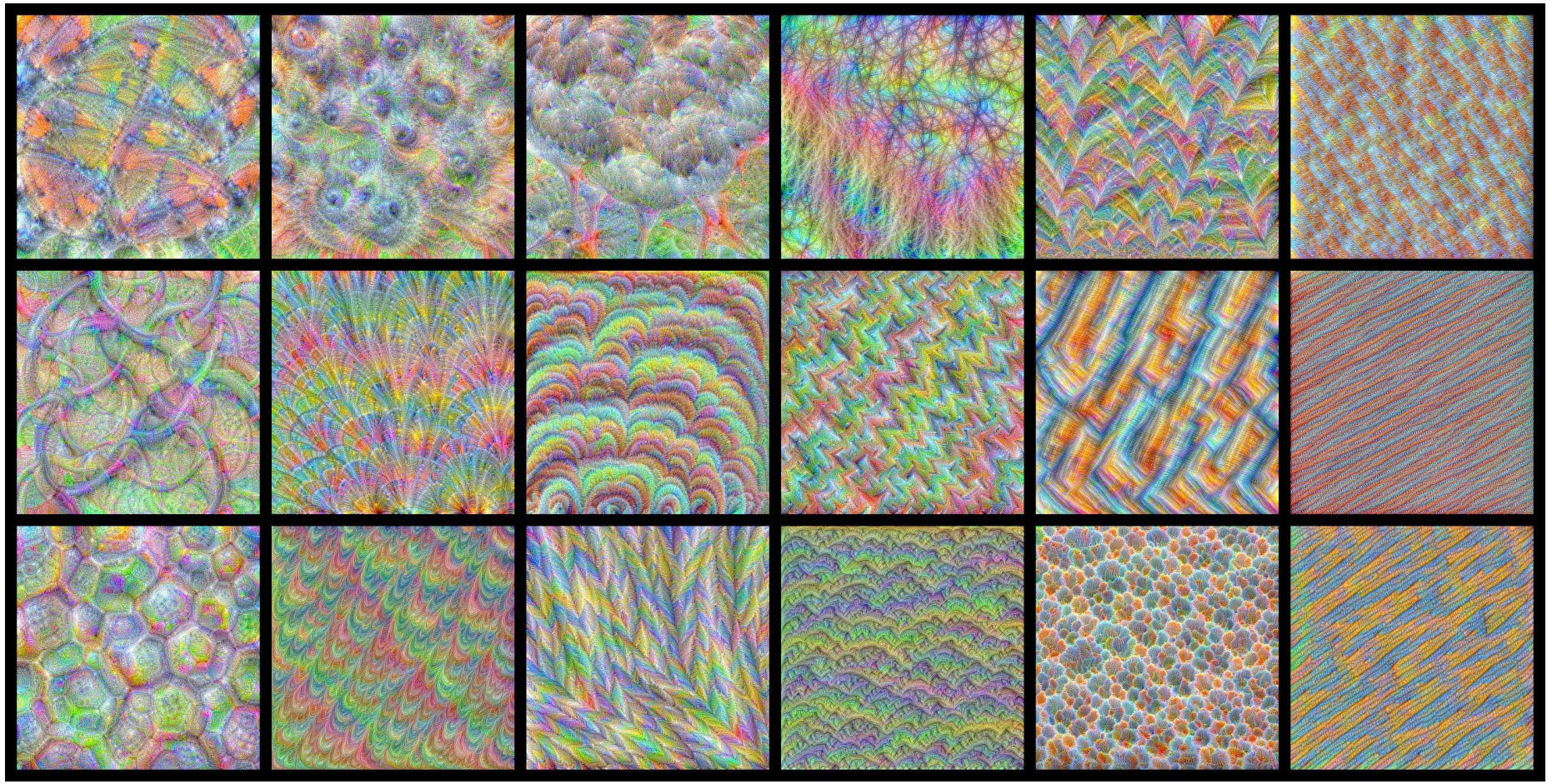


Gradient descent

$$w_j^{(k+1)} = w_j^{(k)} - \eta \left. \frac{\partial E}{\partial \boldsymbol{w}} \right|_{\boldsymbol{w}=w_j^{(k)}}$$



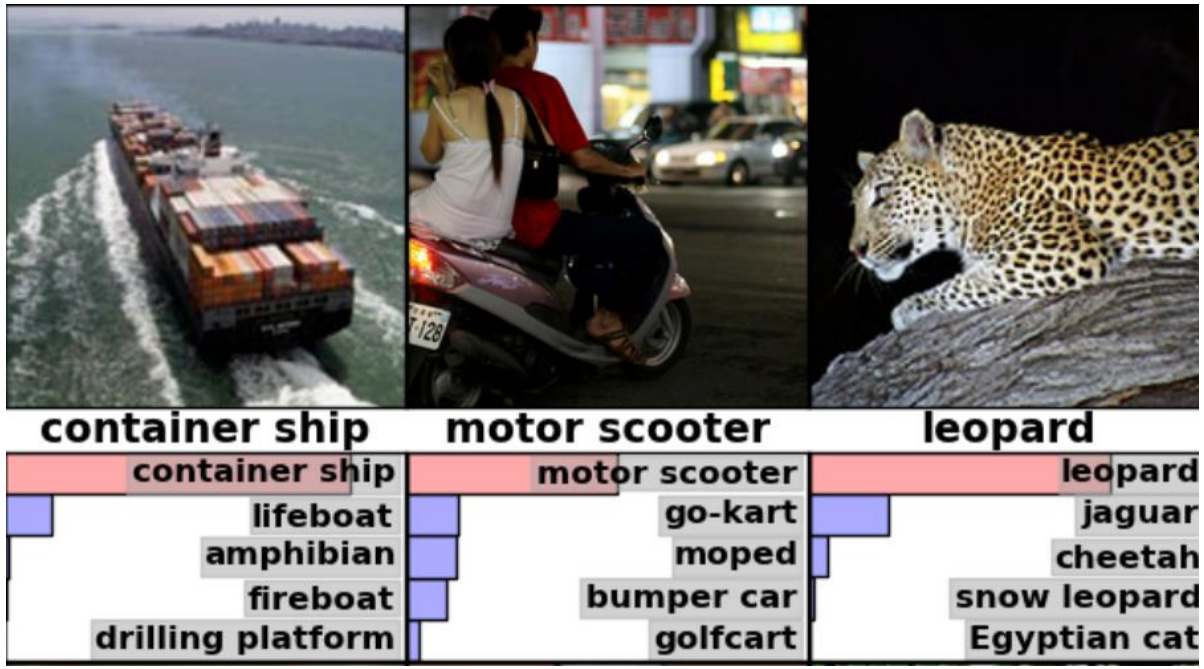
Deep Learning advances



Deep Learning advances

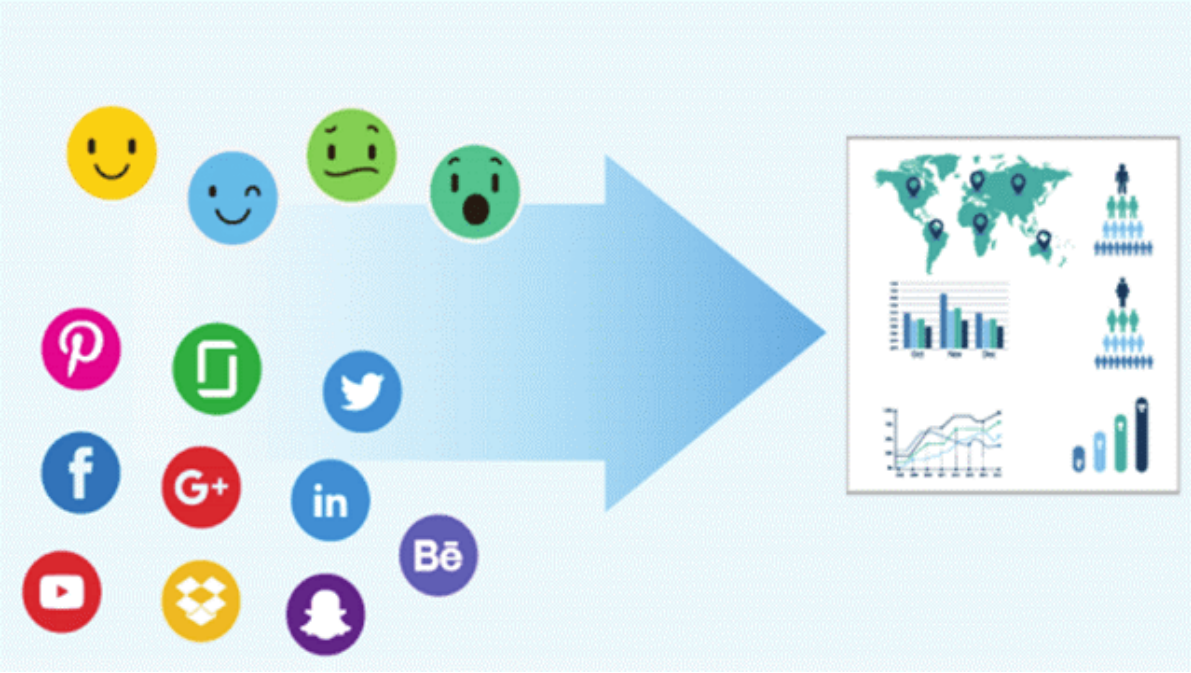
Semantic segmentation

Image recognition



Deep Learning advances

Big data analysis



Decision making

Catch

Explainability

Reconstruct the reasoning that brought the network to a decision, in a human-understandable sense

Data efficiency

In order to train a Deep Neural Network we need a huge dataset

Network initialization

How to initialize the network weights?

[Gilpin et al. 2018]

1. Deep Learning

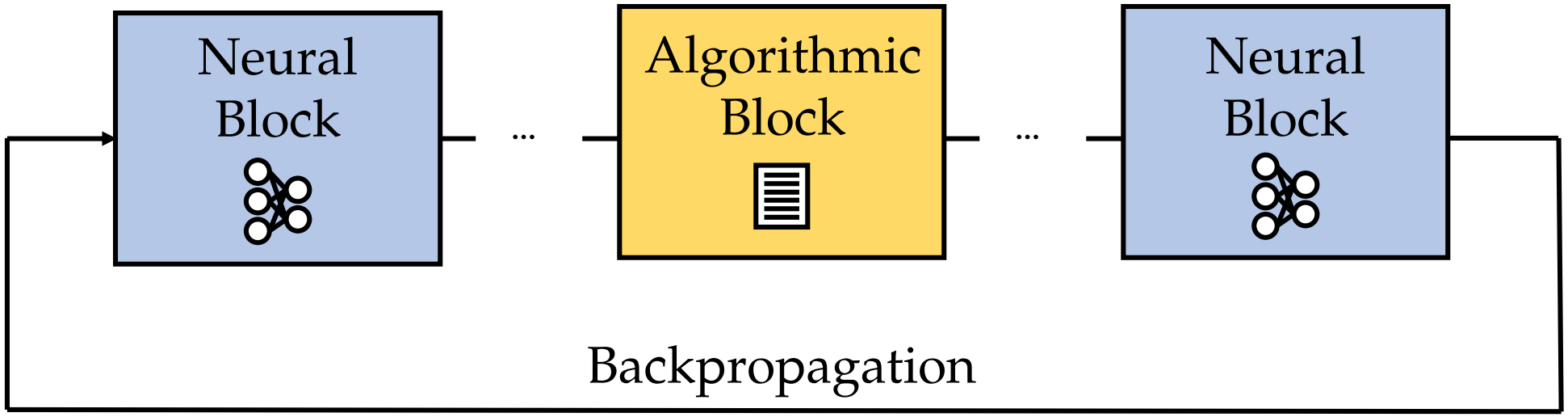
2. Structured Learning

3. Autonomous Navigation

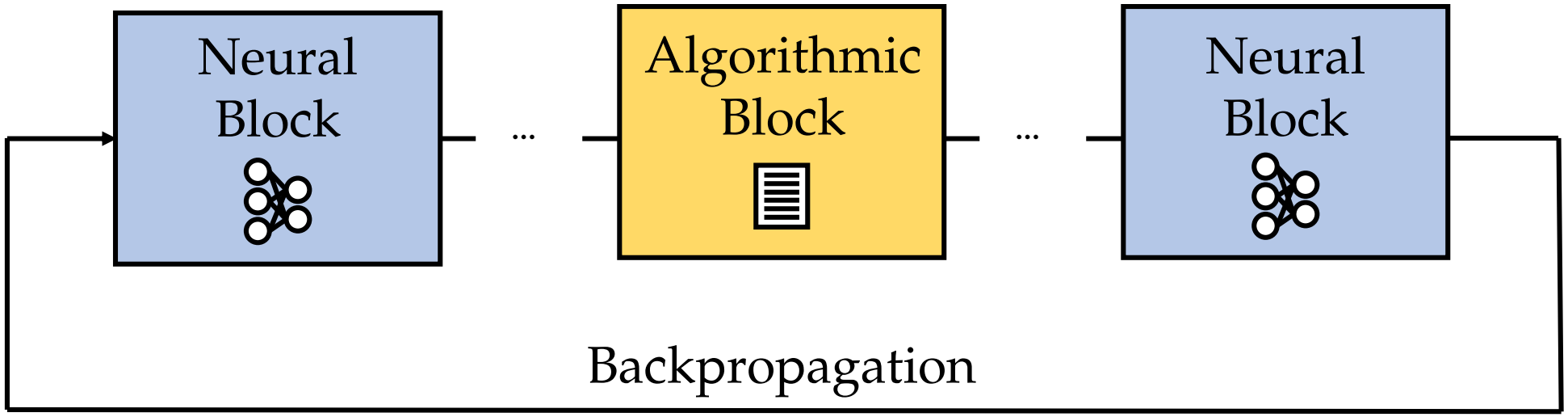
Automatic Differentiation,

Architectures, examples

Algorithmic prior embedding

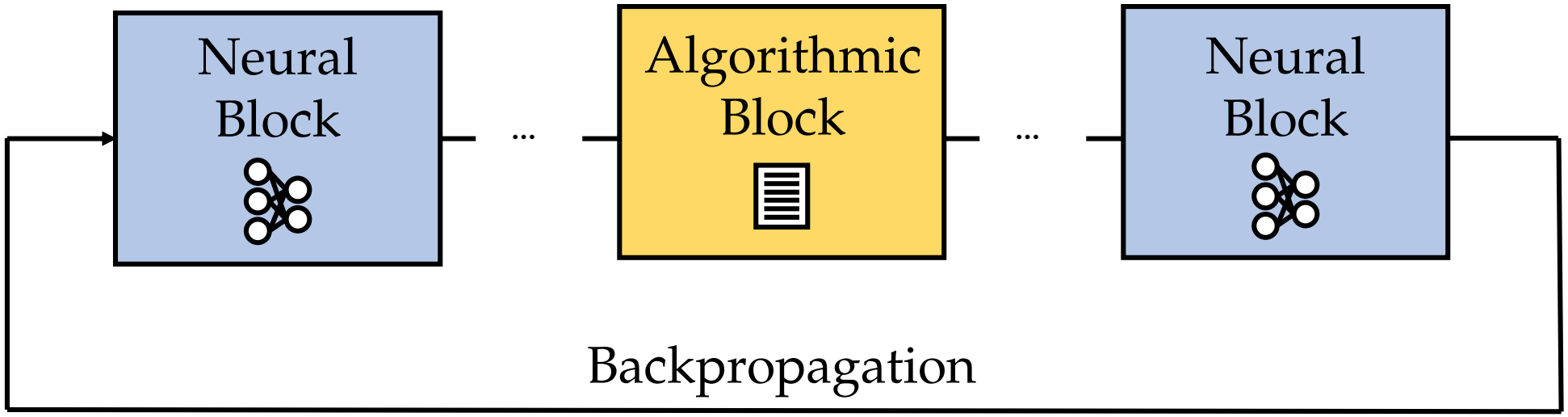


Algorithmic prior embedding



+ End-to-end training

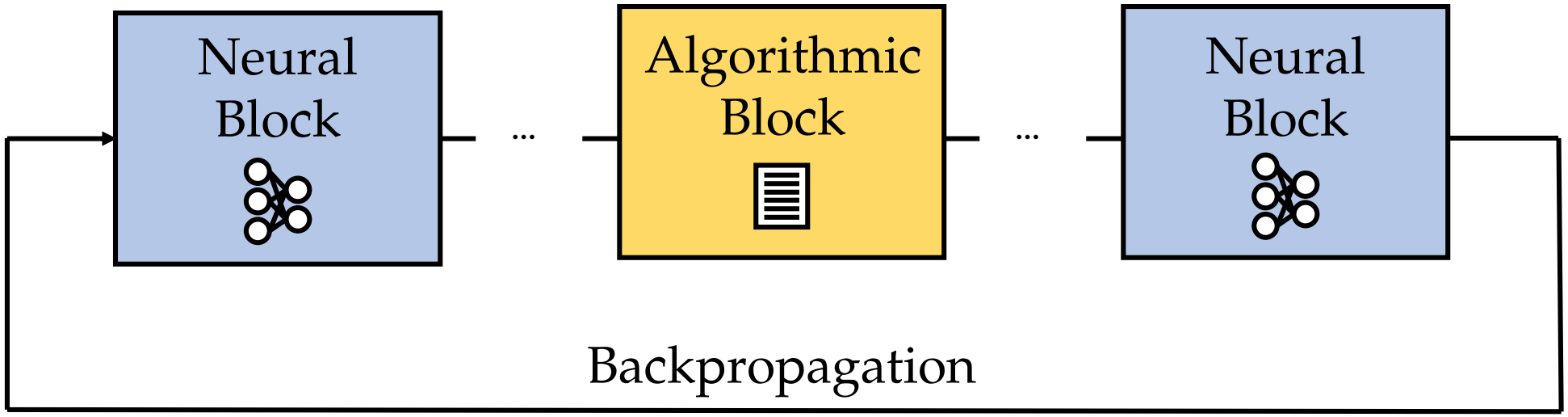
Algorithmic prior embedding



+ End-to-end training

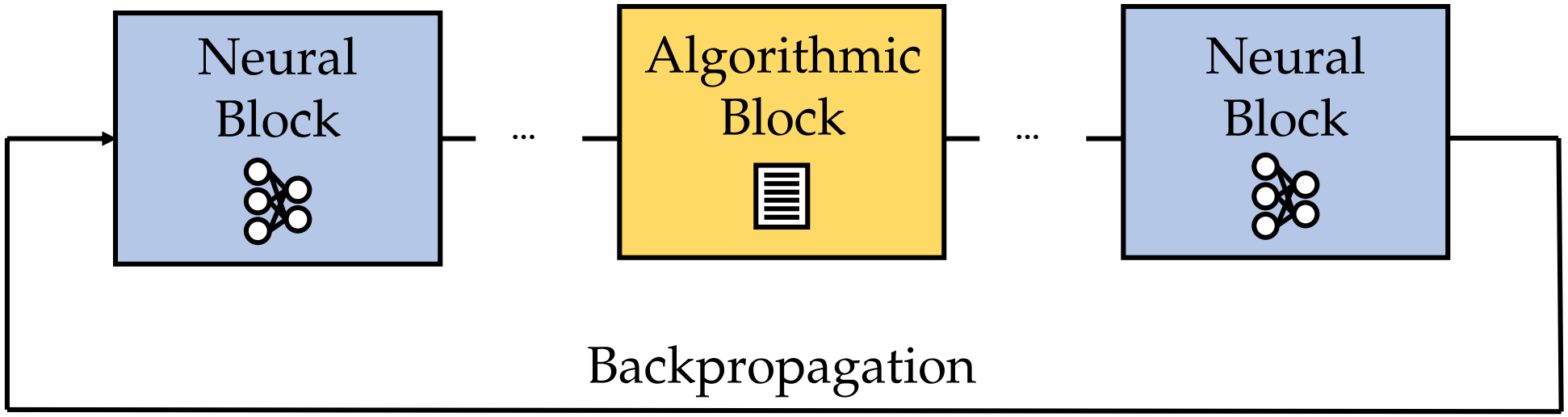
+ Structured solution

Algorithmic prior embedding



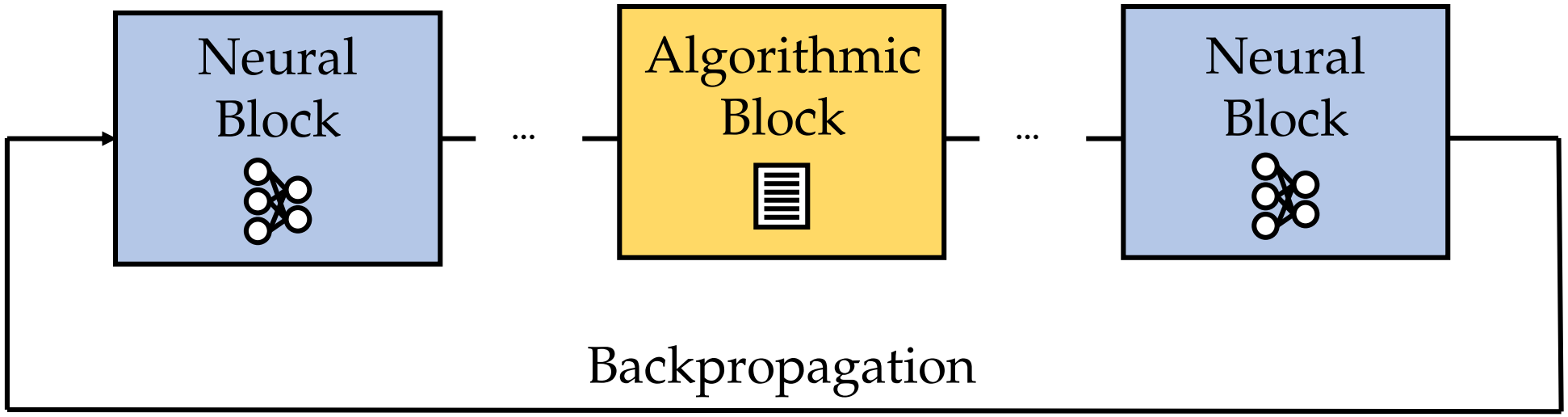
+ End-to-end training
+ Explainability
+ Structured solution

Algorithmic prior embedding



+ End-to-end training
+ Explainability
+ Structured solution
+ Reduce variance

Algorithmic prior embedding

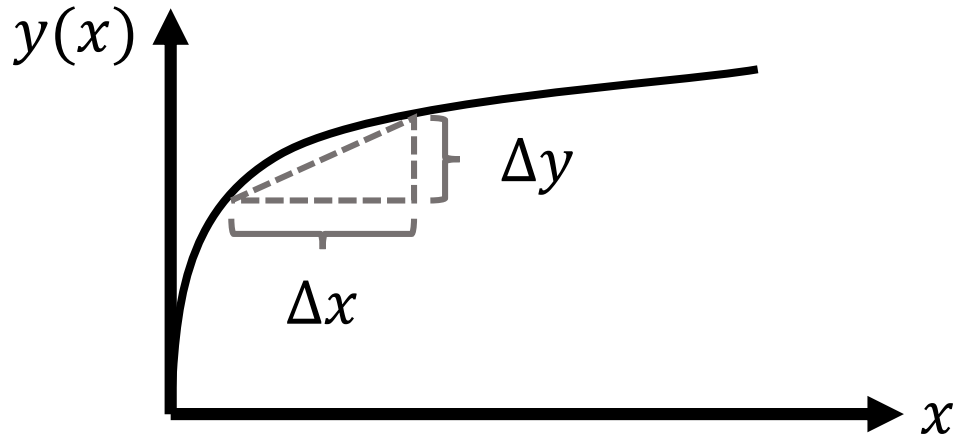


+ End-to-end training
+ Explainability
+ Structured solution
+ Reduce variance

However, the algorithmic block must be differentiable

Derivatives in Math

Differentiation	Description
Numerical	Finite difference approximation
Pros	**Cons**
Easy to implement	Severe approximation errors



[Baydin et al. 2015, Margossian 2018]

Derivatives in Math

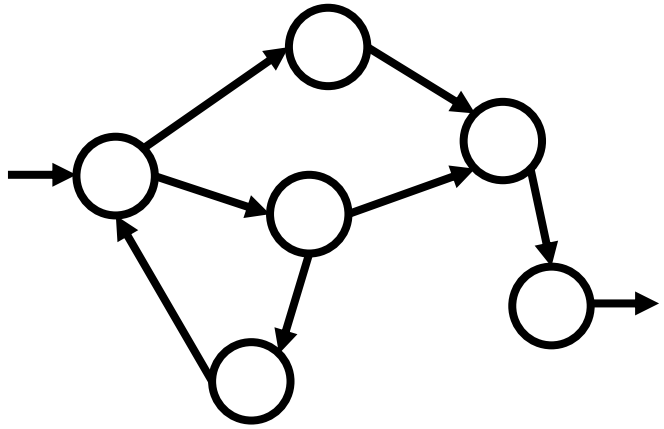
Differentiation	Description
Symbolic	Automatic manipulation of expressions
Pros	**Cons**
Exact solution	Evaluation cost can be exponential

$$
\begin{aligned}
x + y &\rightarrow \dot{x} + \dot{y} \\
x \cdot y &\rightarrow \dot{x} \cdot y + x \cdot \dot{y} \\
\sin(x) &\rightarrow \cos(x) \\
\cos(x) &\rightarrow -\sin(x) \\
&\rightarrow \dots
\end{aligned}
$$

[Baydin et al. 2015, Margossian 2018]

Derivatives in Math

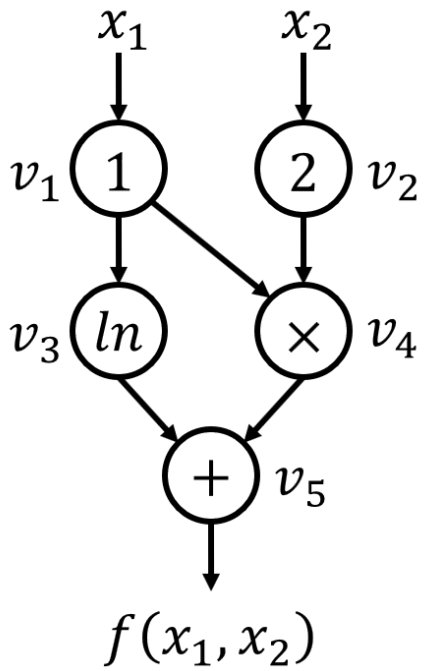
Differentiation	Description
Automatic	Chain rule through a computational graph
Pros	**Cons**
Exact and efficient solution	Hard to implement



[Baydin et al. 2015, Margossian 2018]

Automatic Differentiation

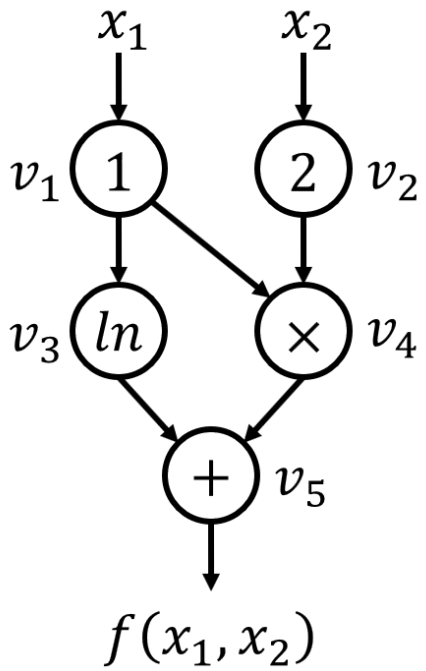
Consider $f(x_1, x_2) = \ln(x_1) + 2x_1x_2$ and evaluate $\left.\frac{\partial f}{\partial x_1}\right|_{(3,4)}$:



v_1	$\dot{v}_1$
v_2	$\dot{v}_2$
v_3	$\dot{v}_3$
v_4	$\dot{v}_4$
v_5	$\dot{v}_5$

Automatic Differentiation

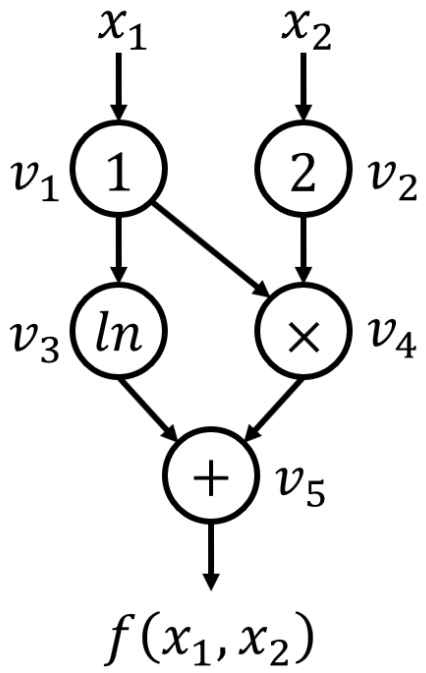
Consider $f(x_1, x_2) = \ln(x_1) + 2x_1x_2$ and evaluate $\left.\frac{\partial f}{\partial x_1}\right|_{(3,4)}$:



$v_1 = x_1 = 3$	$\dot{v}_1$
$v_2 = 2x_2 = 8$	$\dot{v}_2$
$v_3 = \ln(v_1) = 1.1$	$\dot{v}_3$
$v_4 = v_1 \cdot v_2 = 24$	$\dot{v}_4$
$v_5 = v_3 + v_4 = 25.1$	$\dot{v}_5$

Automatic Differentiation

Consider $f(x_1, x_2) = \ln(x_1) + 2x_1x_2$ and evaluate $\left.\frac{\partial f}{\partial x_1}\right|_{(3,4)}$:



$v_1 = x_1 = 3$	$\dot{v}_1 = 1$
$v_2 = 2x_2 = 8$	$\dot{v}_2 = 0$
$v_3 = \ln(v_1) = 1.1$	$\dot{v}_3 = \frac{\dot{v}_1}{v_1} = 0.33$
$v_4 = v_1 \cdot v_2 = 24$	$\dot{v}_4 = \dot{v}_1 \cdot v_2 + v_1\dot{v}_2 = 8$
$v_5 = v_3 + v_4 = 25.1$	$\dot{v}_5 = \dot{v}_3 + \dot{v}_4 = 8.33$

Automatic Differentiation

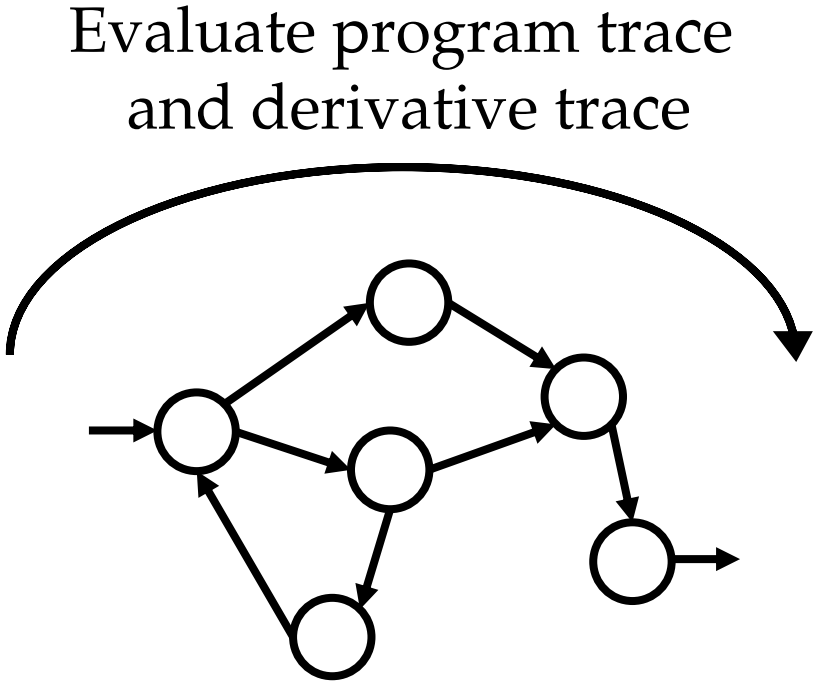
- Collection of techniques to evaluate derivatives of numeric functions, expressed as computer programs
- Any code of numerical manipulation eventually results in an evaluation trace over which the chain rule can be applied

[Baydin et al. 2015, Margossian 2018]

Automatic Differentiation

Forward mode

- Implemented with dual numbers and operator overloading
- Intuitive
- Convenient with multiple outputs

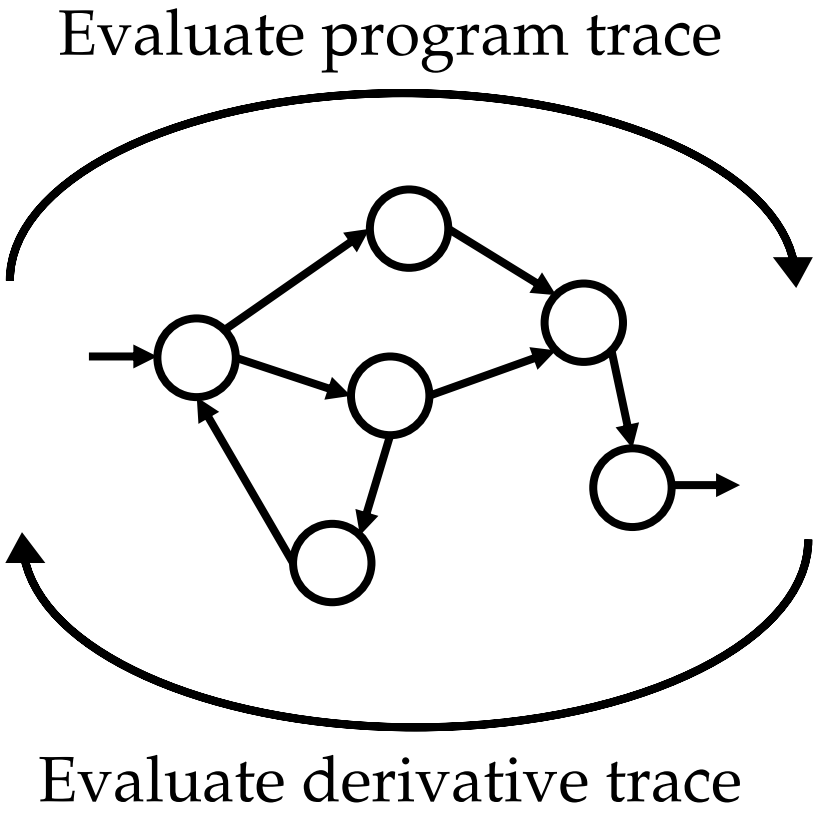


[Baydin et al. 2015, Margossian 2018]

Automatic Differentiation

Backward mode

- Generalized backpropagation
- Requires more memory than Forward mode
- Convenient with multiple inputs



[Baydin et al. 2015, Margossian 2018]

Issues

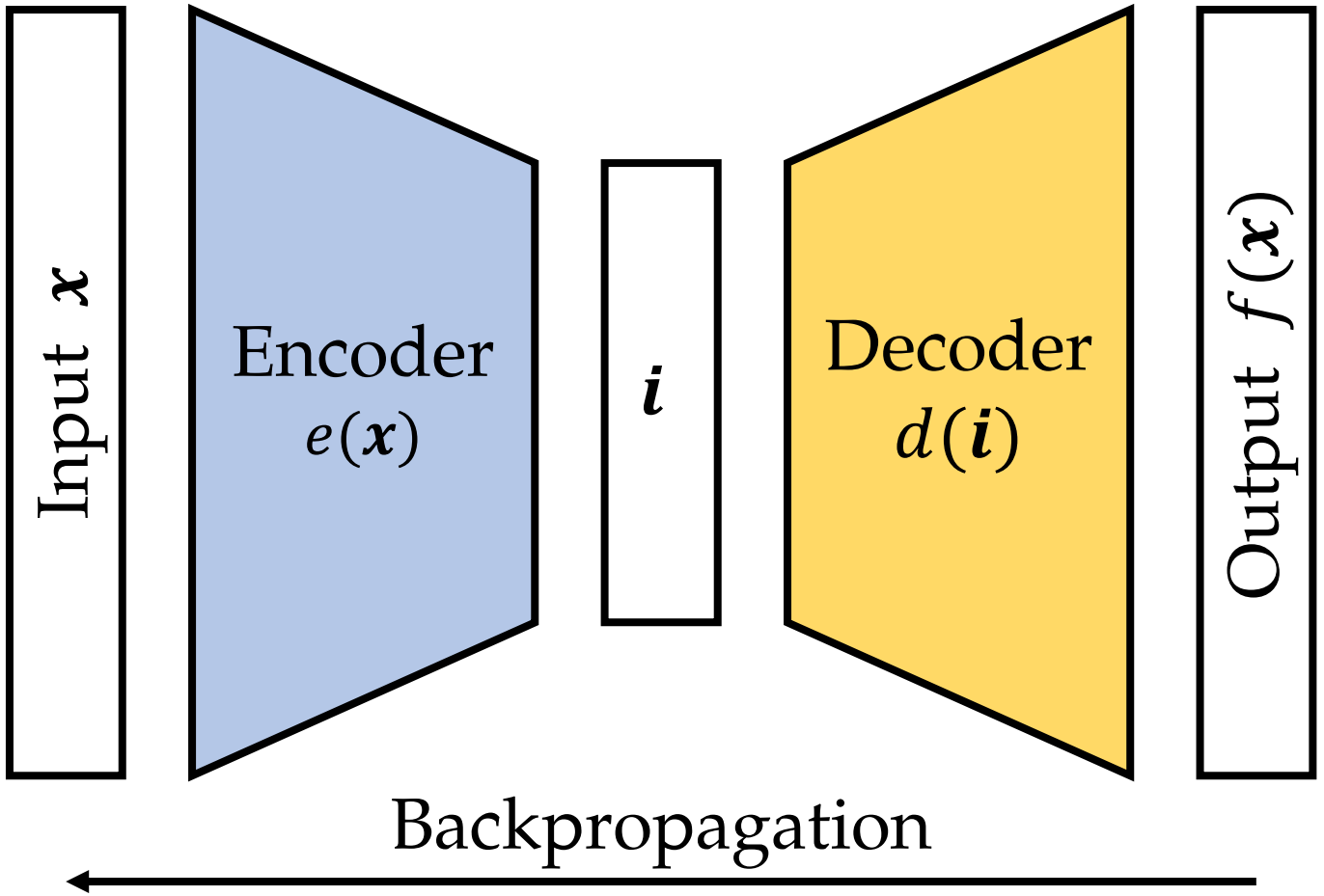
Consider the program

```
Bad-AD(x):
    if (x == 3):
        return 9
    else:
        return x * x
```

What about the derivative at x=3?

It should be 6, but AD returns 0.

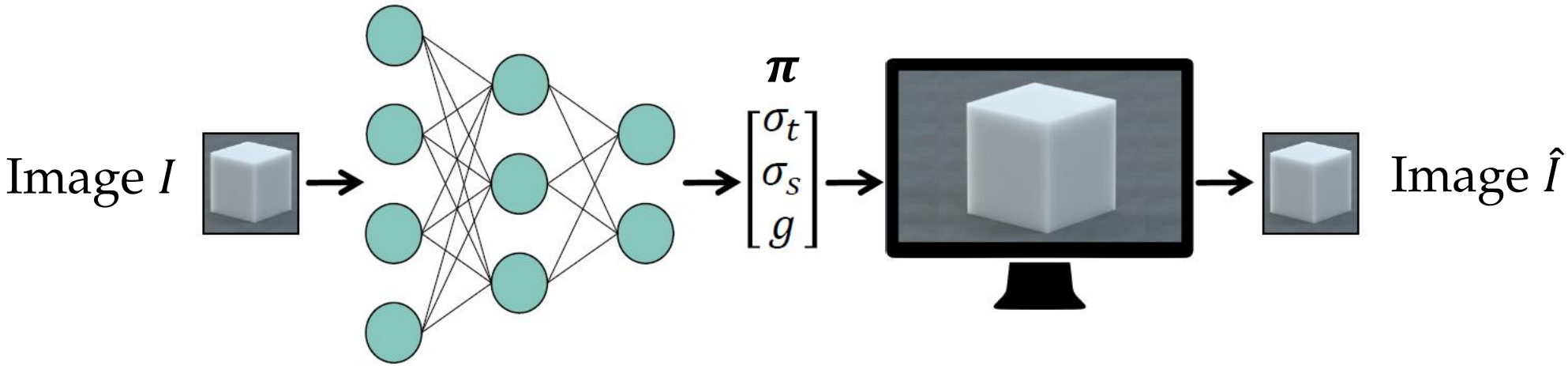
Structured Autoencoder



● : Neural Network

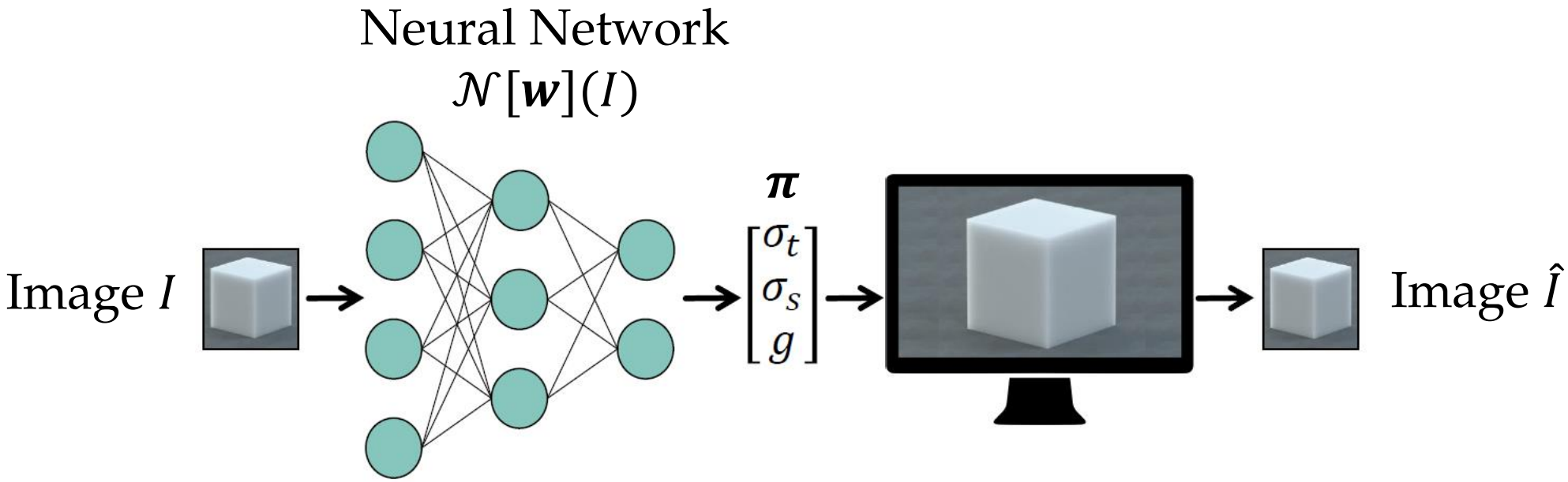
● : Differentiable algorithm

Structured Autoencoder



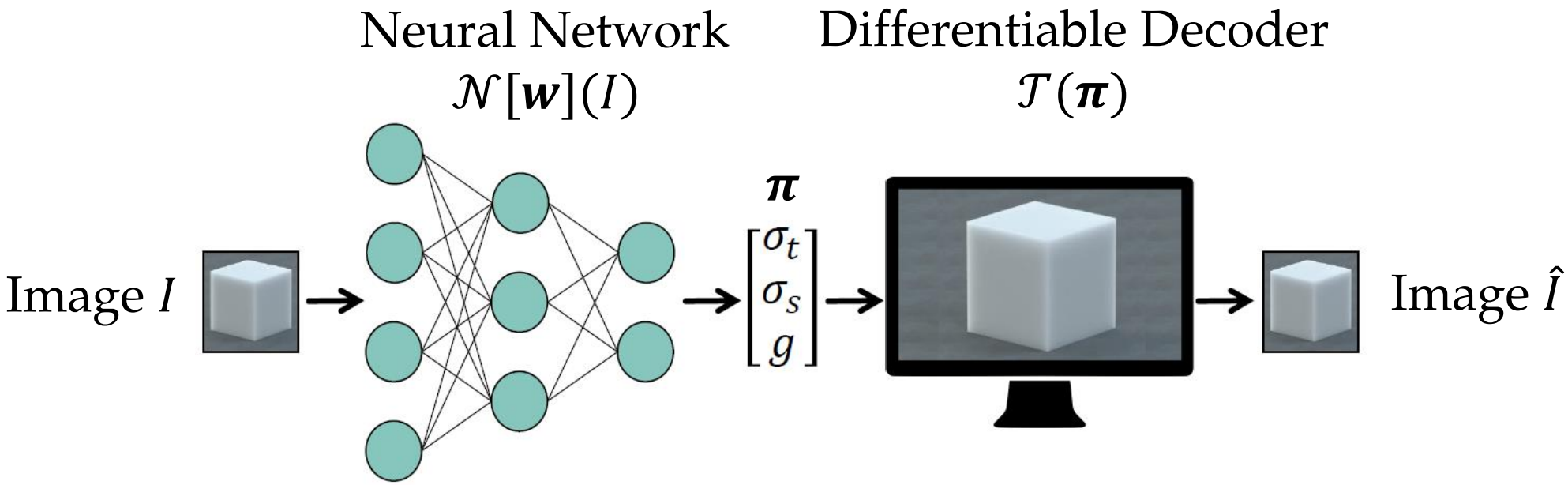
[Che et al. 2018]

Structured Autoencoder



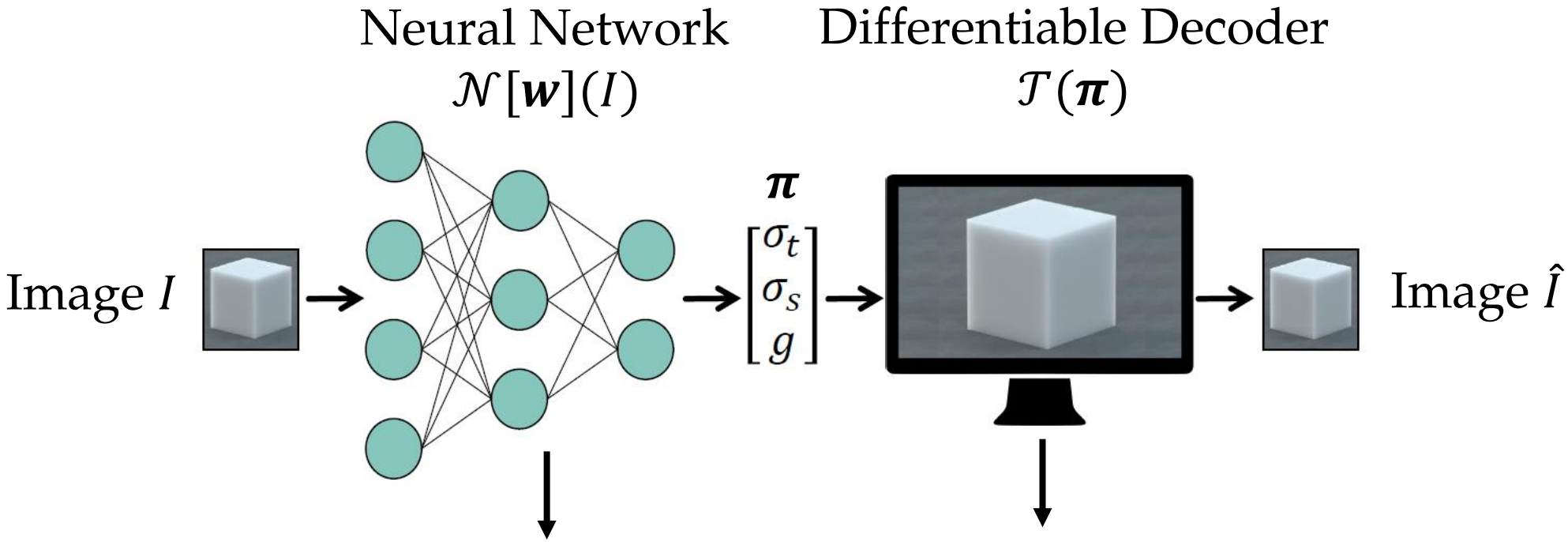
[Che et al. 2018]

Structured Autoencoder



[Che et al. 2018]

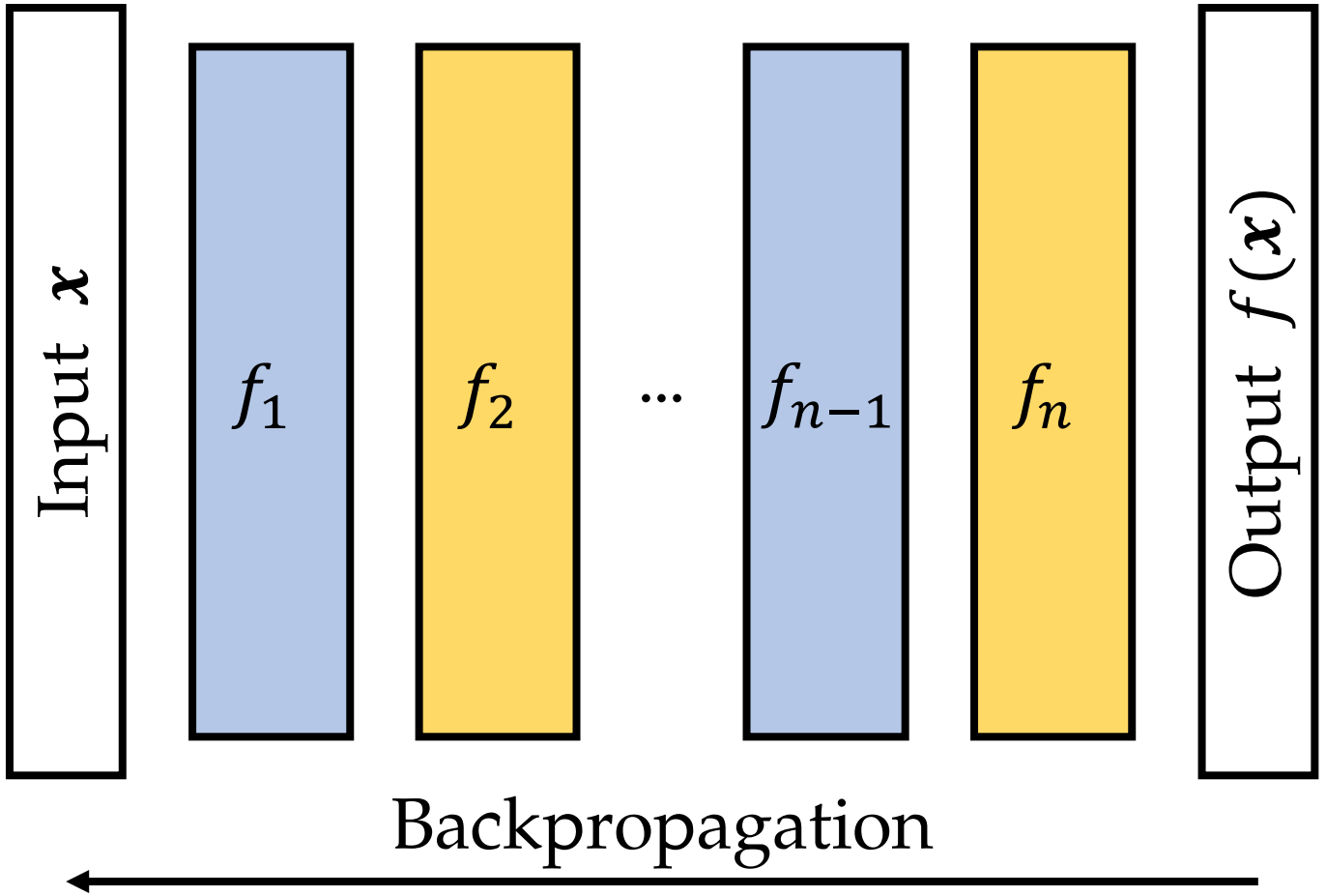
Structured Autoencoder



$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \underbrace{\|\boldsymbol{\pi} - \mathcal{N}[\boldsymbol{w}](I)\|^2}_{\text{Supervised loss}} + \lambda \underbrace{\|I - \mathcal{T}(\mathcal{N}[\boldsymbol{w}](I))\|^2}_{\text{Regularization}}$$

[Che et al. 2018]

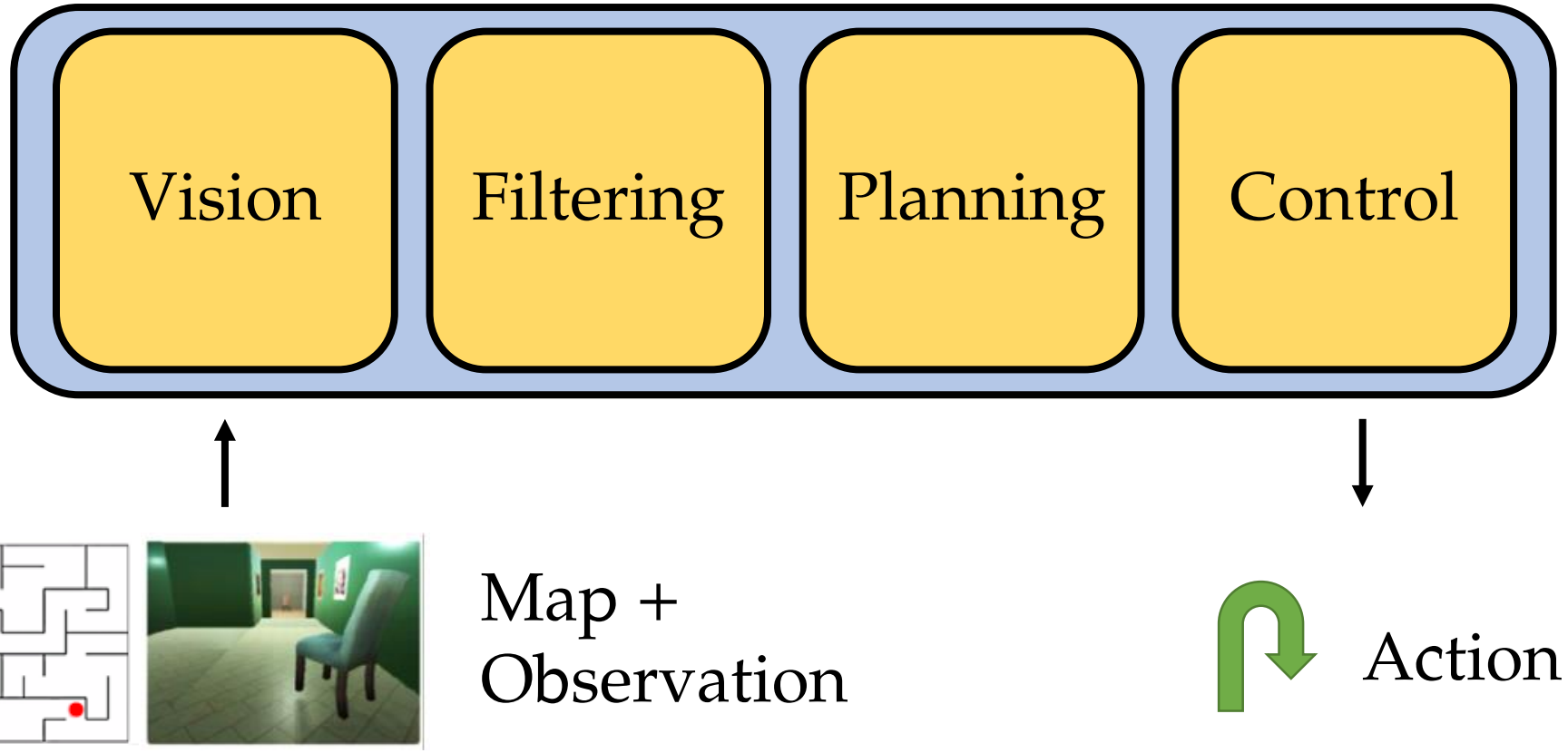
Structured Pipeline



● : Neural Network

● : Differentiable algorithm

Structured Pipeline



[Karkus et al. 2019]

1. Deep Learning
2. Structured Learning
3. Autonomous Navigation

Advantages of SL integration,
our contribution, research plan

Research and industry

- Prior knowledge for object detection, planning, control, multiagent coordination, etc.
- Autonomous driving:
  - Increase explainability
  - Model verifiability
  - Lower data requirements

[Badue et al. 2019, Grigorescu et al. 2019]

Our contribution

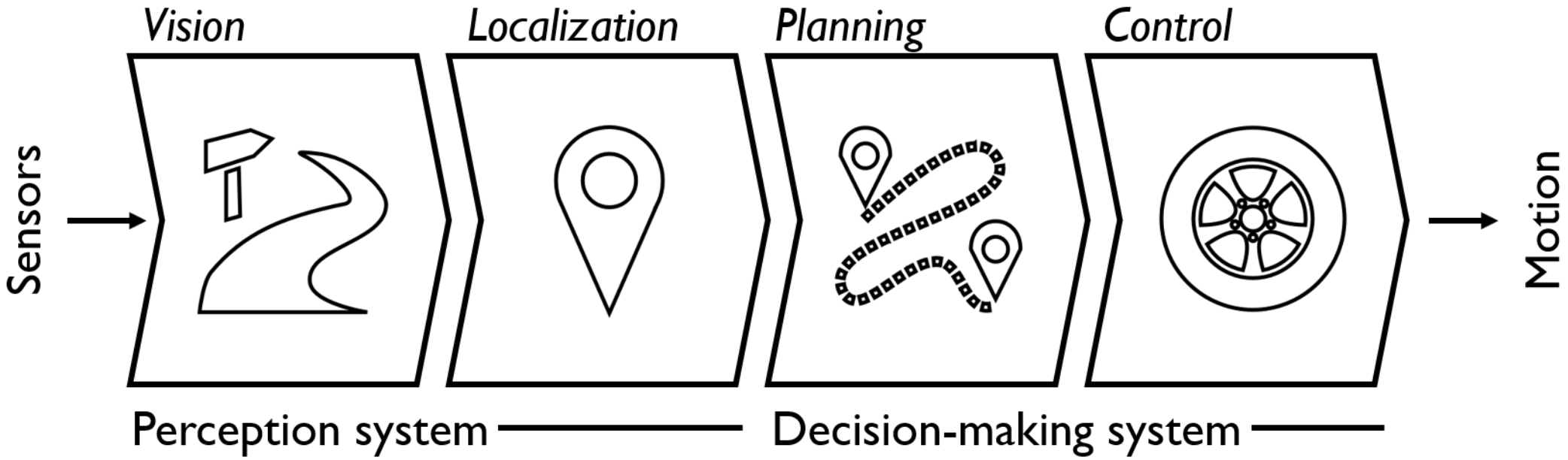
- Show the potential of Structured Learning in a complex simulation
- Build a Structured Pipeline in order to empirically test the strengths and weaknesses of Structured Learning at scale
- Evaluate explainability in fault scenarios
- Implement the system in a real robotic agent

1. Differentiation inquiry

Analyze and compare AD-based techniques with non-AD-based techniques for algorithmic differentiation

	AD-based techniques	Non-AD-based techniques
Pros	Strong technological support and research results	Less technological support, few research efforts
Cons	Boundary conditions must be handled	Provides a derivative everywhere

2. Architecture design



[Karkus et al. 2019]

3. Implementation and training

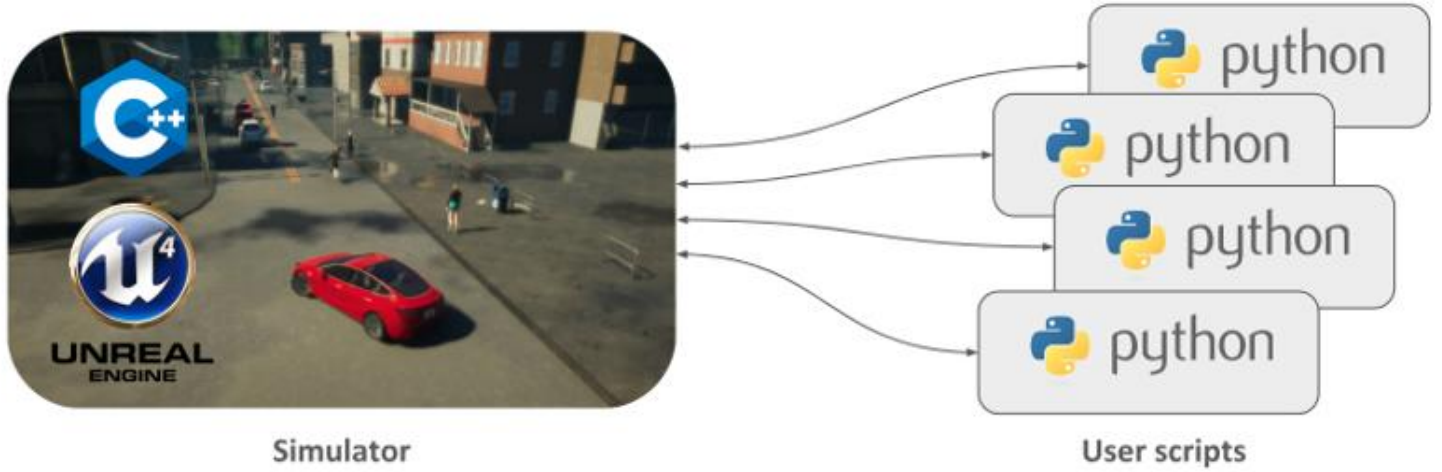
PYTORCH

Deep Learning with PyTorch

Choose a suitable ML framework, given differentiation and architectural requirements

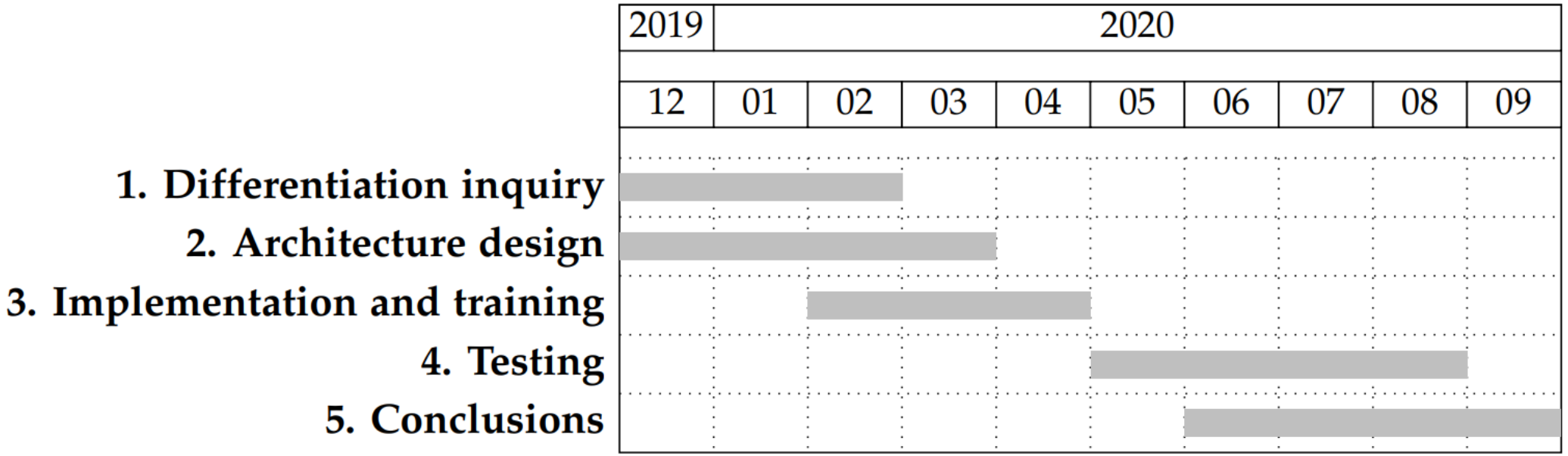
TensorFlow 2.0

CARLA



4. Testing and Conclusions

- Evaluate performance with respect to the current standards (explainability, accuracy, sample efficiency)
- Write the final paper



References

- GILPIN ET AL. Explaining explanations: An overview of interpretability of machine learning, 2018
- BAYDIN ET AL. Automatic differentiation in machine learning: a survey, 2015
- MARGOSSIAN A review of automatic differentiation and its efficient implementation, 2018
- CHE ET AL. Inverse transport networks, 2018
- KARKUS ET AL. Differentiable algorithm networks for composable robot learning, 2019
- BADUE ET AL. Self-driving cars: A survey, 2019
- GRIGORESCU ET AL. A survey of deep learning techniques for autonomous driving, 2019