Modularity as a Means for Complexity Management in Neural Networks Learning

David Castillo-Bolado
Cayetano Guerra-Artal
Mario Hernández-Tejera

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

INSTITUTO UNIVERSITARIO SIANI INGENIERIA COMPUTACIONAL

Contents

Modularity

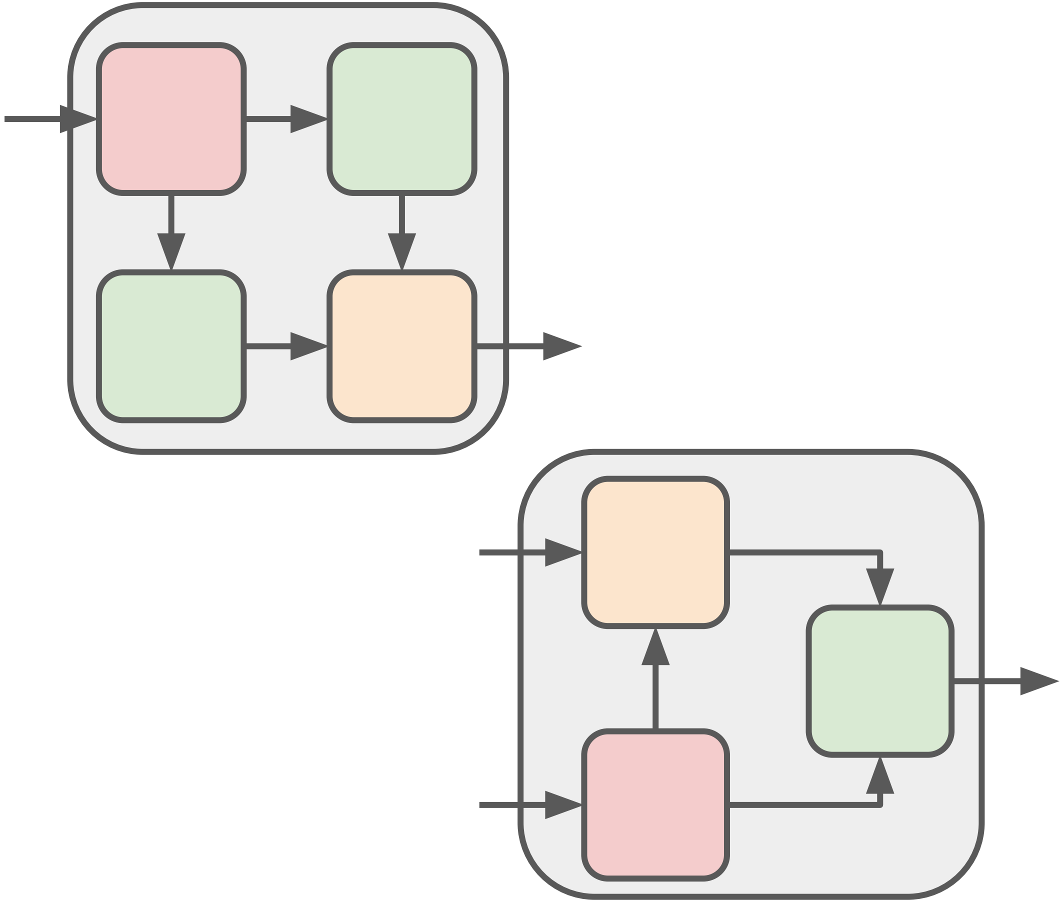
A key concept in Engineering

Modularity

Coupling & Cohesion

Advantages

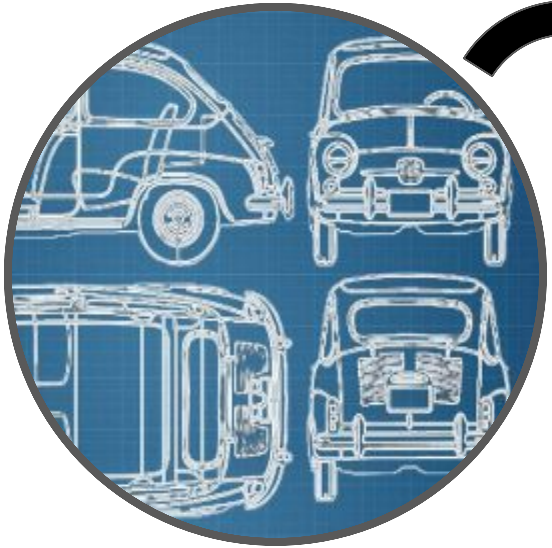
- Abstraction
- Complexity Limiting
- Reutilization



Divide & Conquer

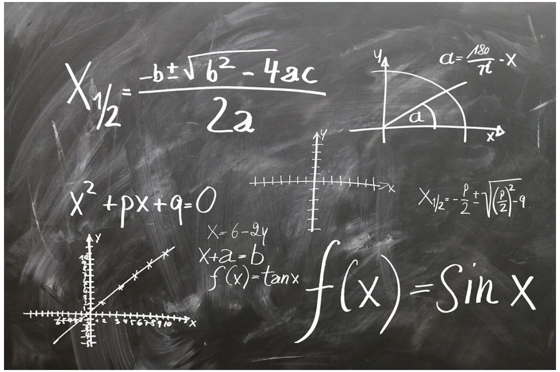
1st decompose

3rd combine solutions



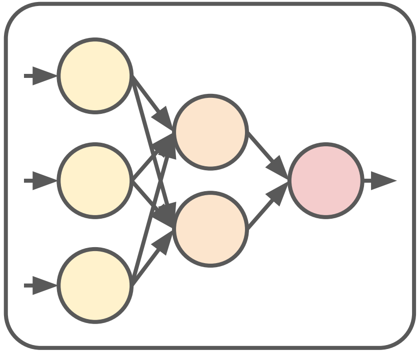
2nd work on subproblems

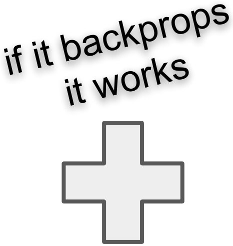
The power of composition



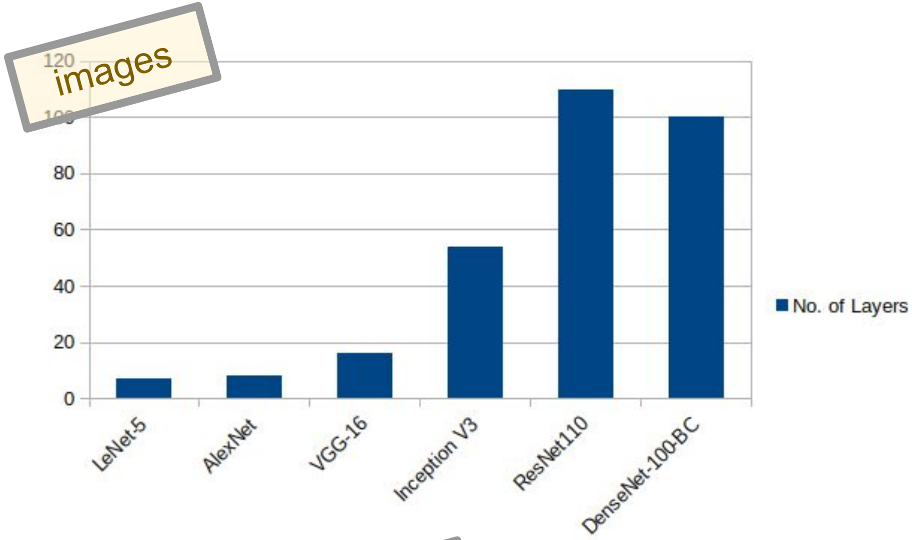
Modularity
In Neural networks

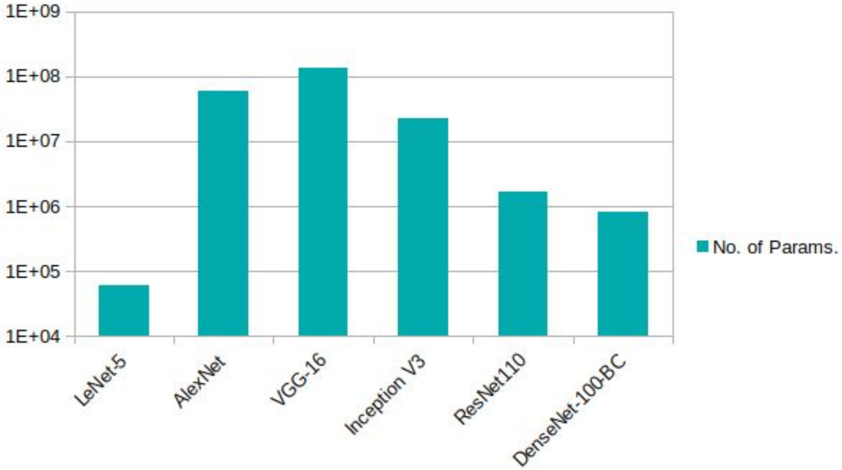
Generic approach to Neural Networks



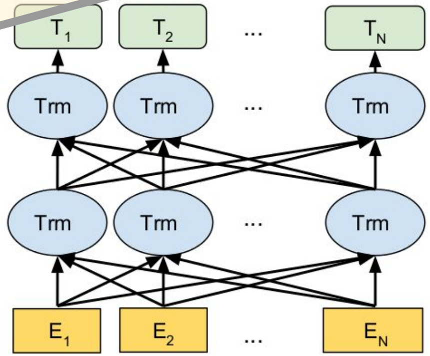


images

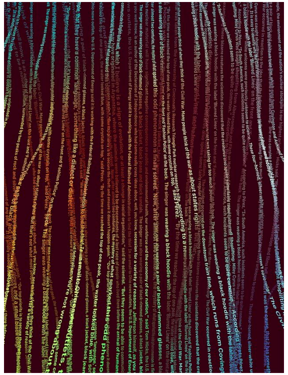




text



$\text{BERT}_{\text{LARGE}}$ - 380M



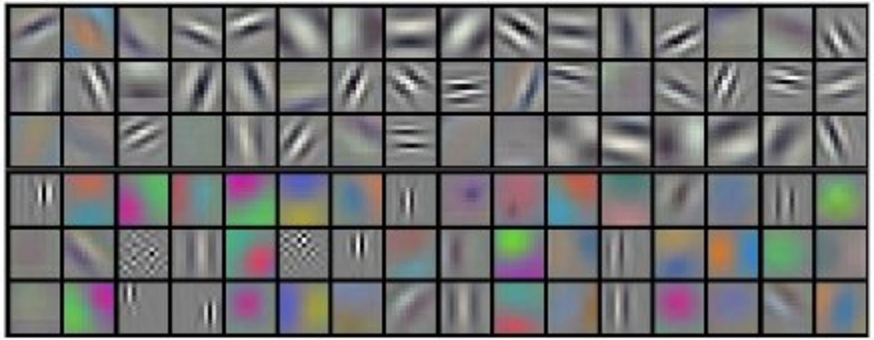
GPT-2 - 1.5G

Modularity in Neural Networks

transfer learning / fine-tuning

word2vec

```
# instantiate the model
model <- application_resnet50(weights = 'imagenet')
```

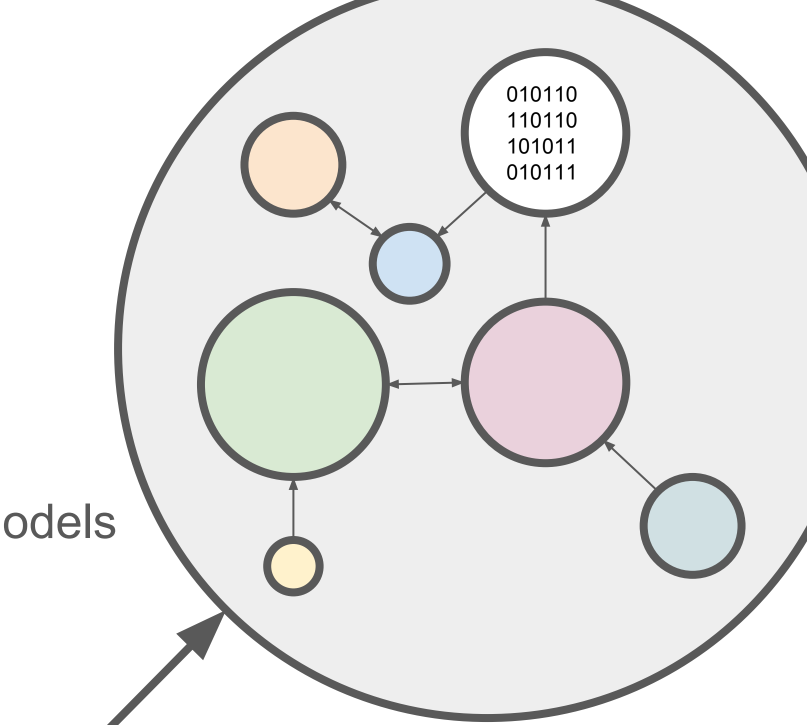


Hypothesis

It is possible to obtain a functional equivalent of a jointly trained artificial neural network by partitioning the model and training the parts individually.

Foreseeable advantages

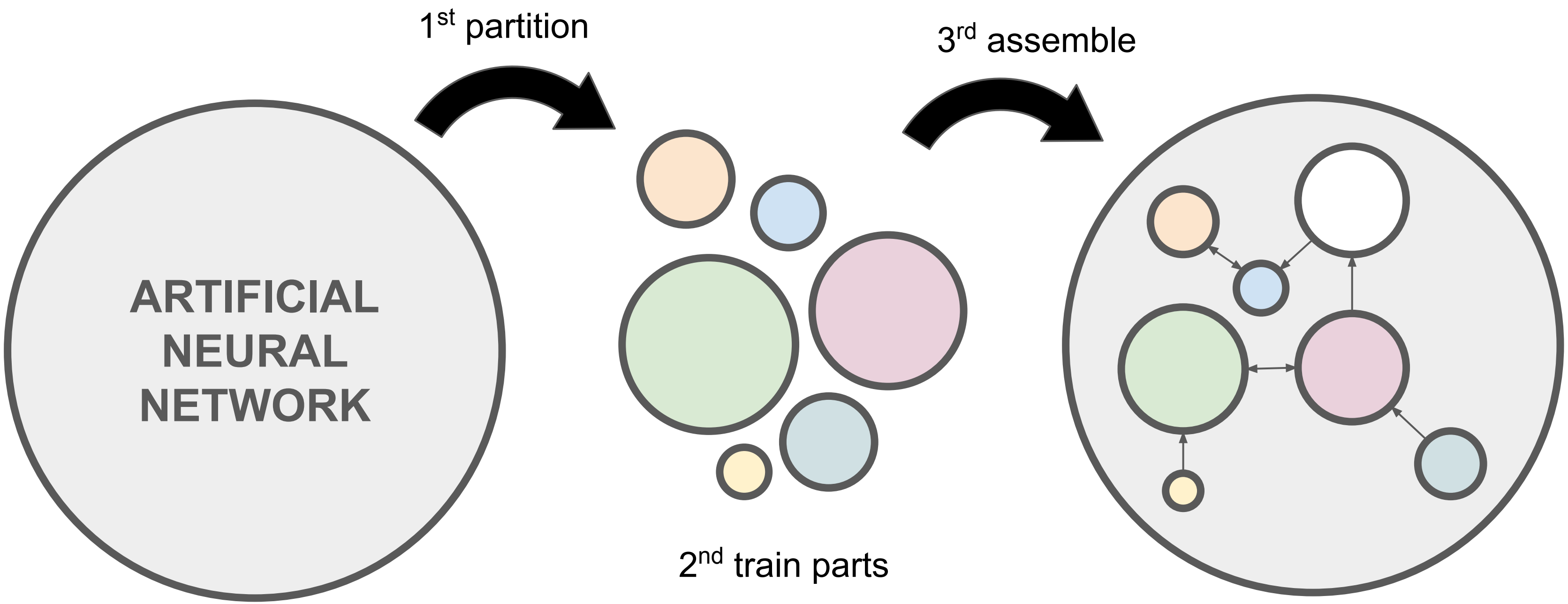
- Exploitation of parallelism at training time
- Reuse of parts among different models
- Tackle complexity by building hierarchical models
- Decoupling of parts



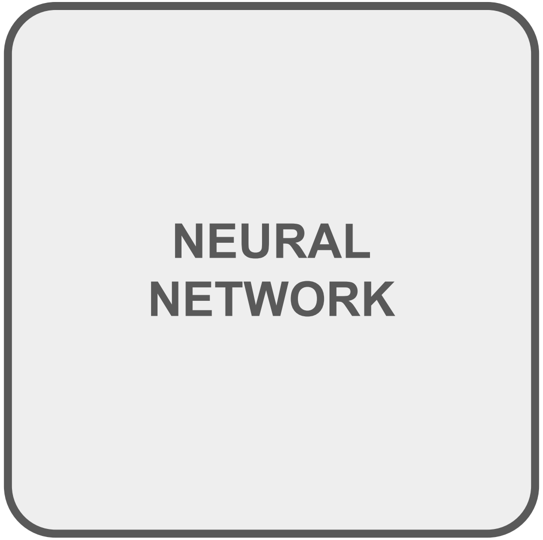
Concerns

- Guarantee functional equivalency

Modular training of Neural Networks



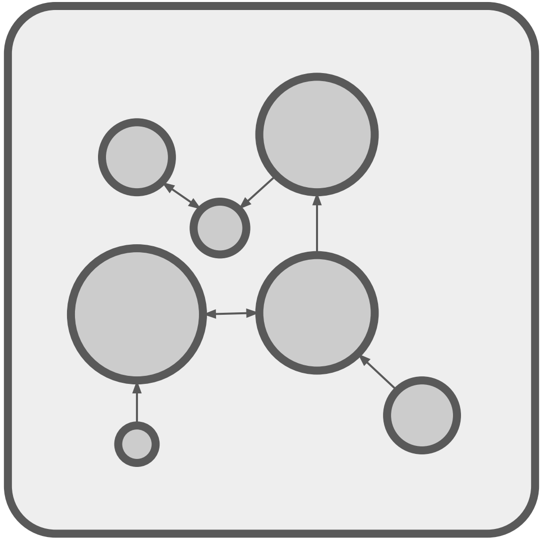
Modular training of Neural Networks



Modular training of Neural Networks

1. Identify modules

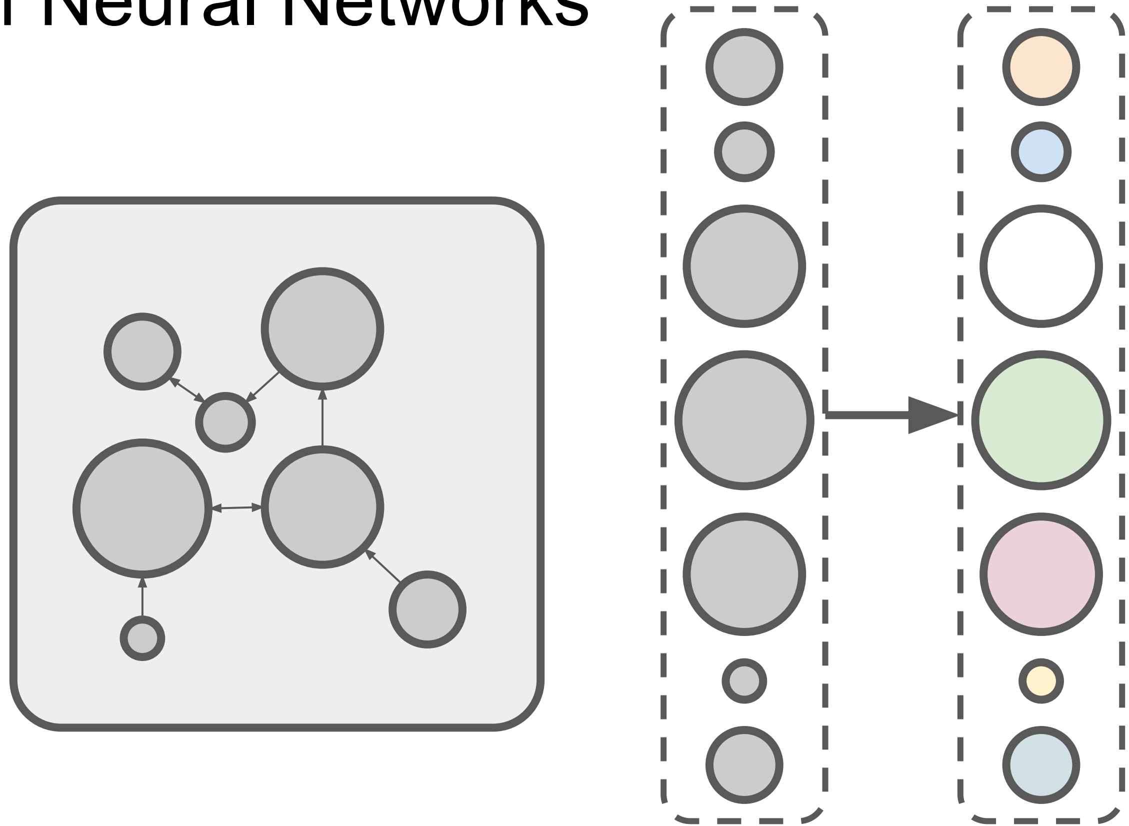
1st knowledge injection



Modular training of Neural Networks

1. Identify modules
1st knowledge injection

2. Train modules
2nd knowledge injection

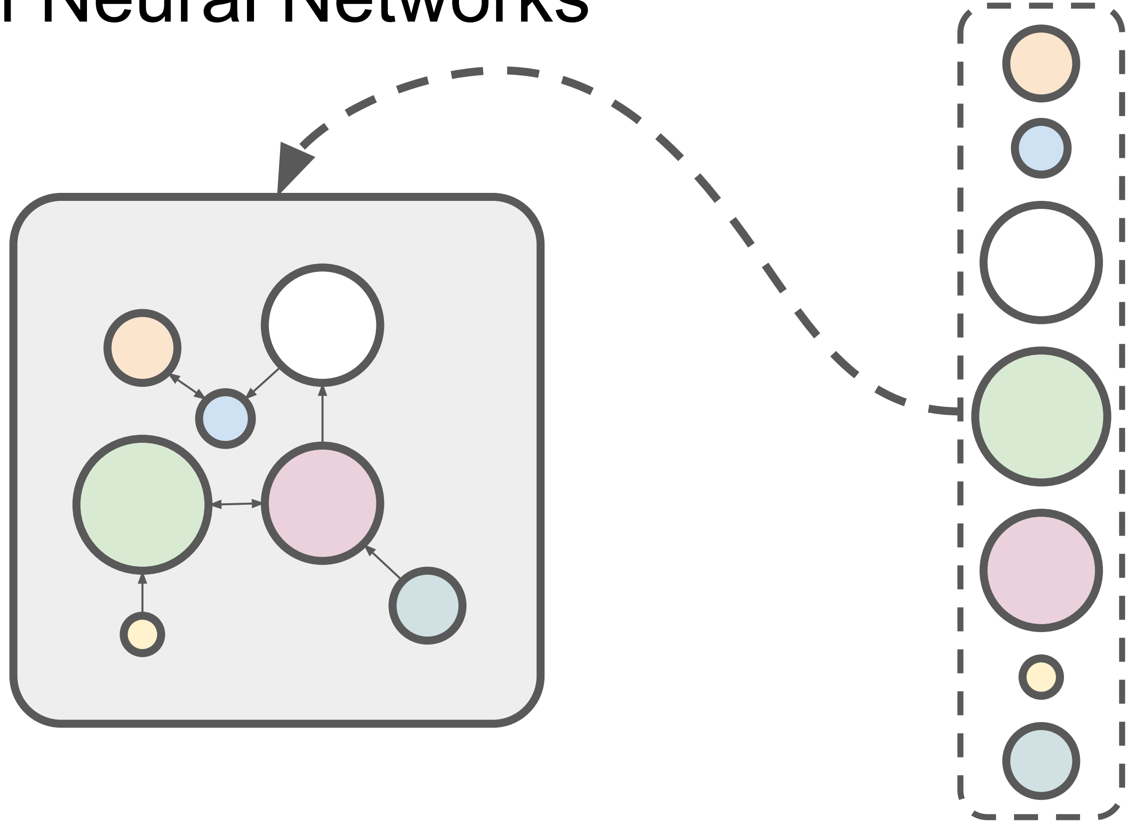


Modular training of Neural Networks

1. Identify modules
1st knowledge injection

2. Train modules
2nd knowledge injection

3. Assemble



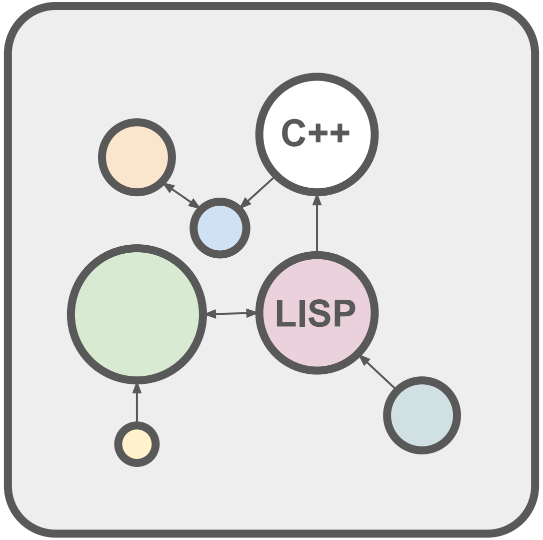
Modular training of Neural Networks

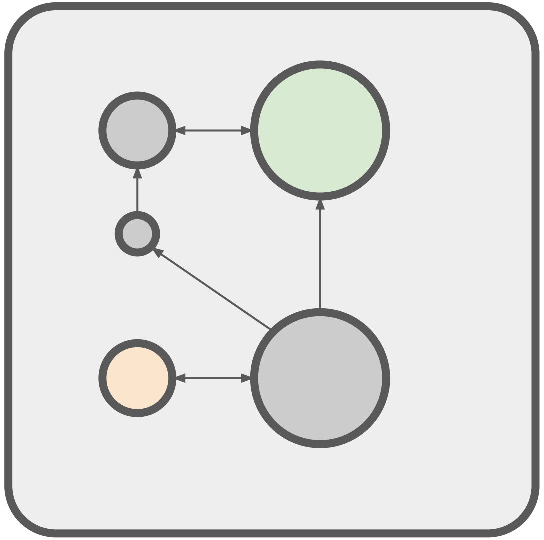
1. Identify modules
1st knowledge injection

2. Train modules
2nd knowledge injection

3. Assemble

4. Reuse or Reimplementation
3rd knowledge injection





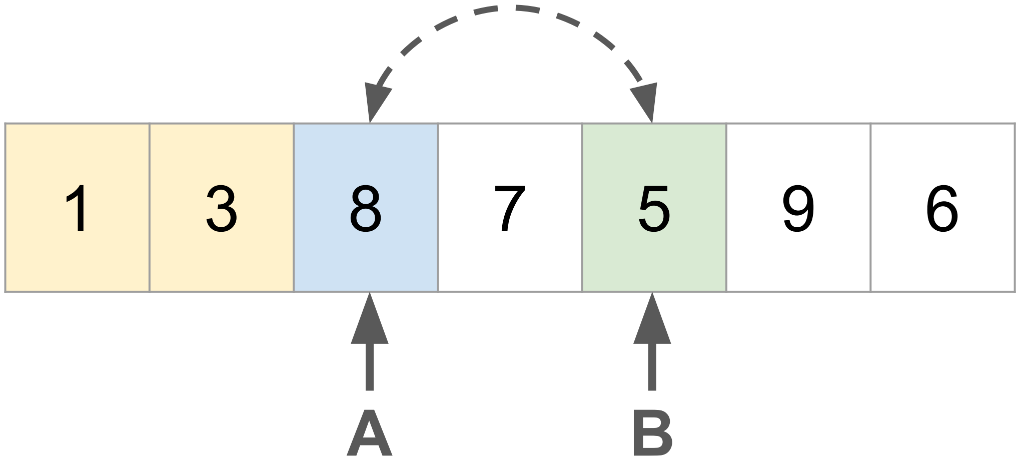
Case Study

The Selection Sort Algorithm

Desiderata

- Simple implementation
- Known primitive operations
- Configurable sample complexity

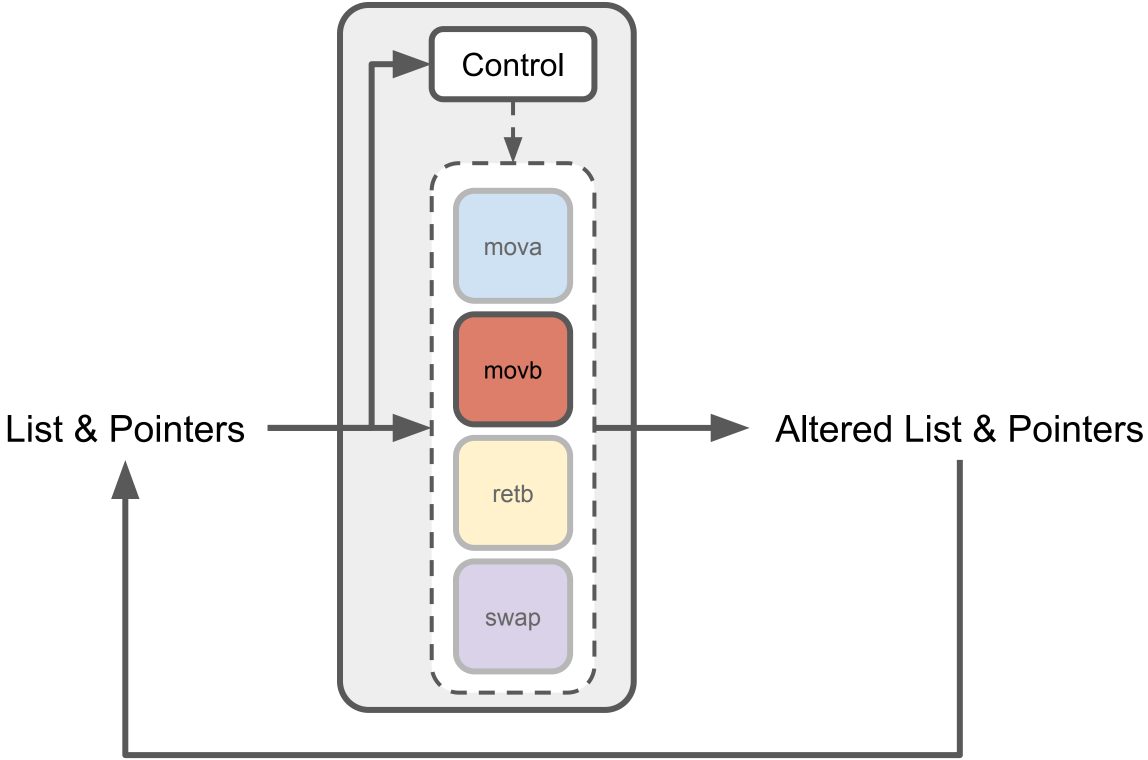
Selection Sort



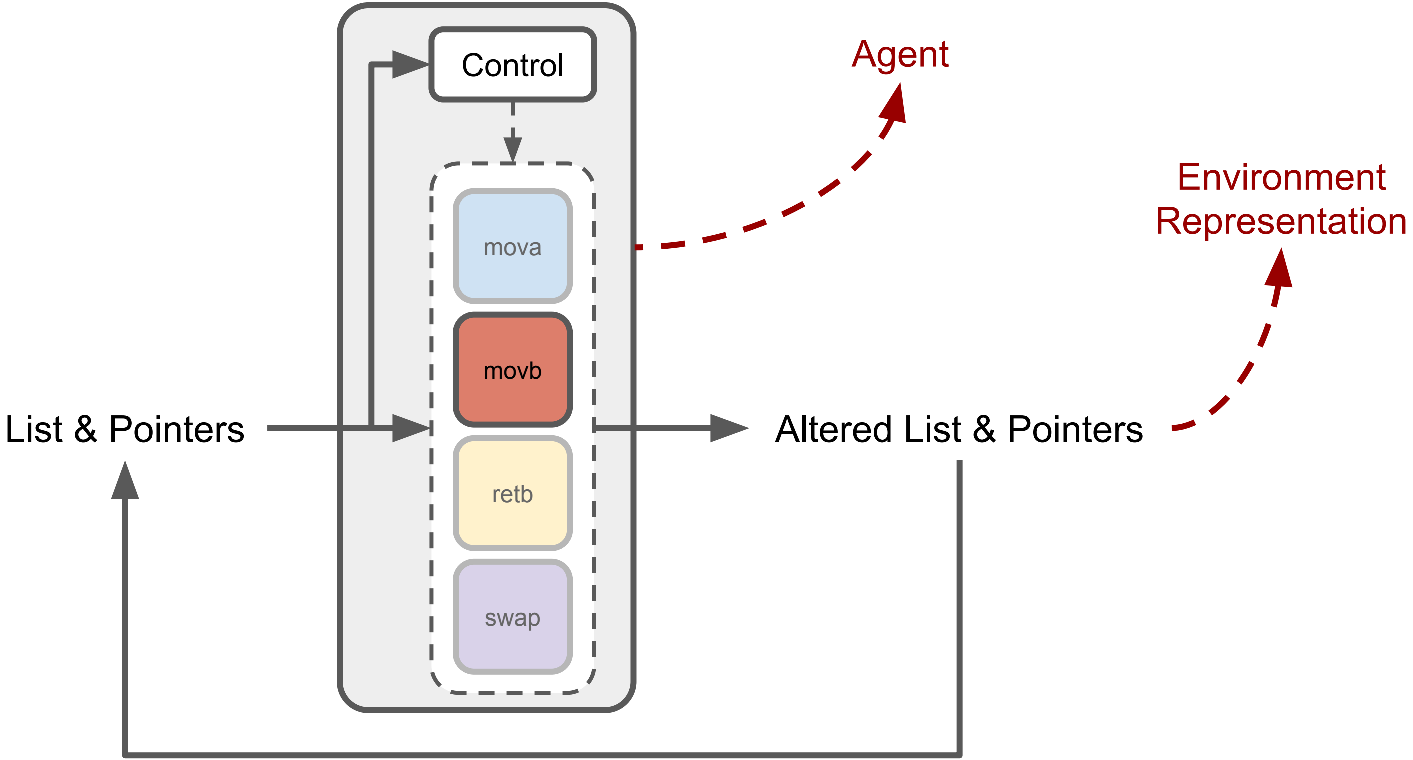
Primitive operations

- **Move A** to the right
- **Move B** to the right
- **Return B** right next to A
- **Swap** values between A and B

Selection Sort: implementation

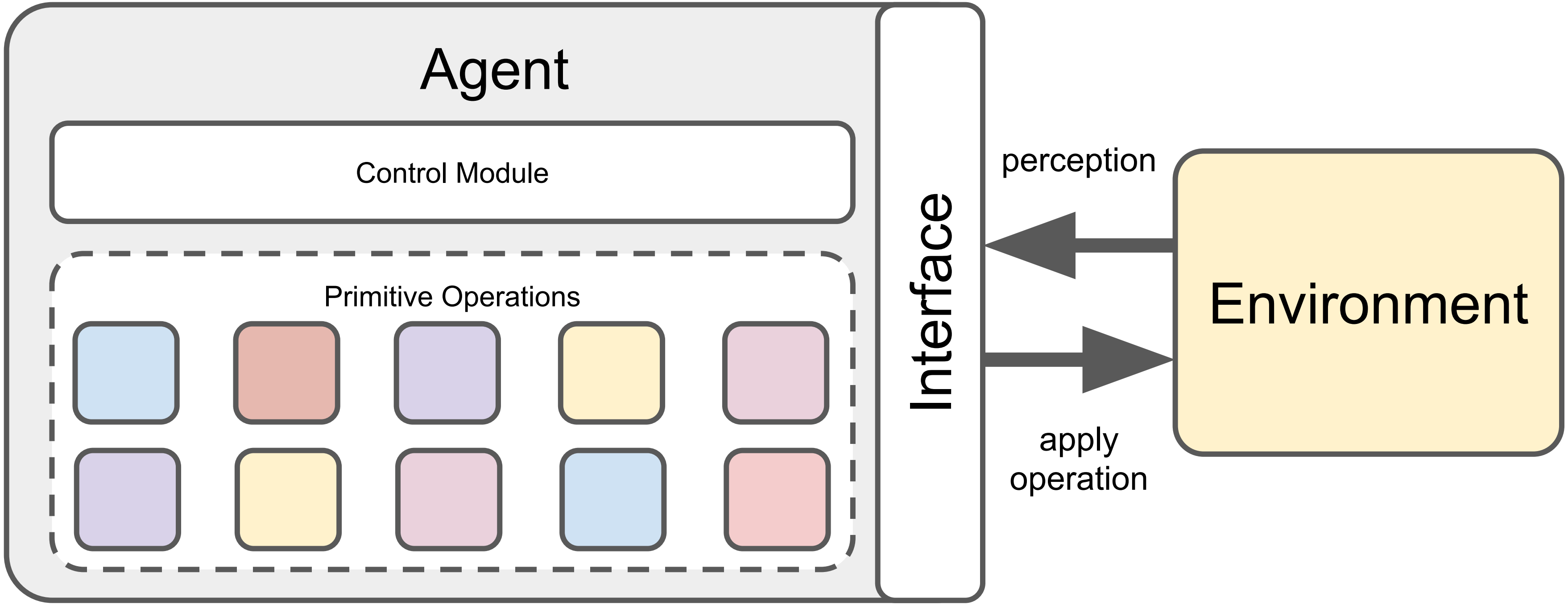


Selection Sort: implementation

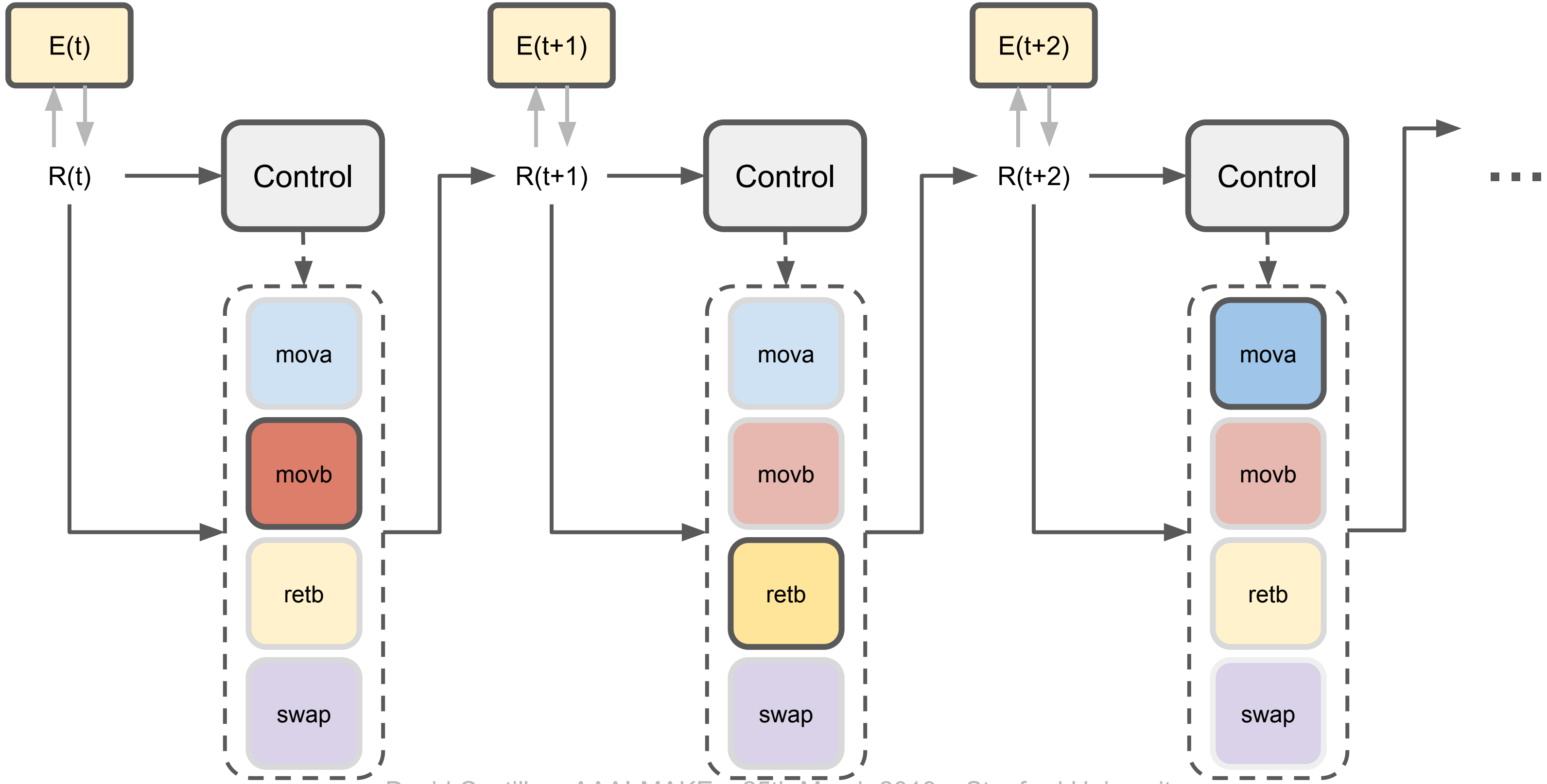


The Modular Architecture

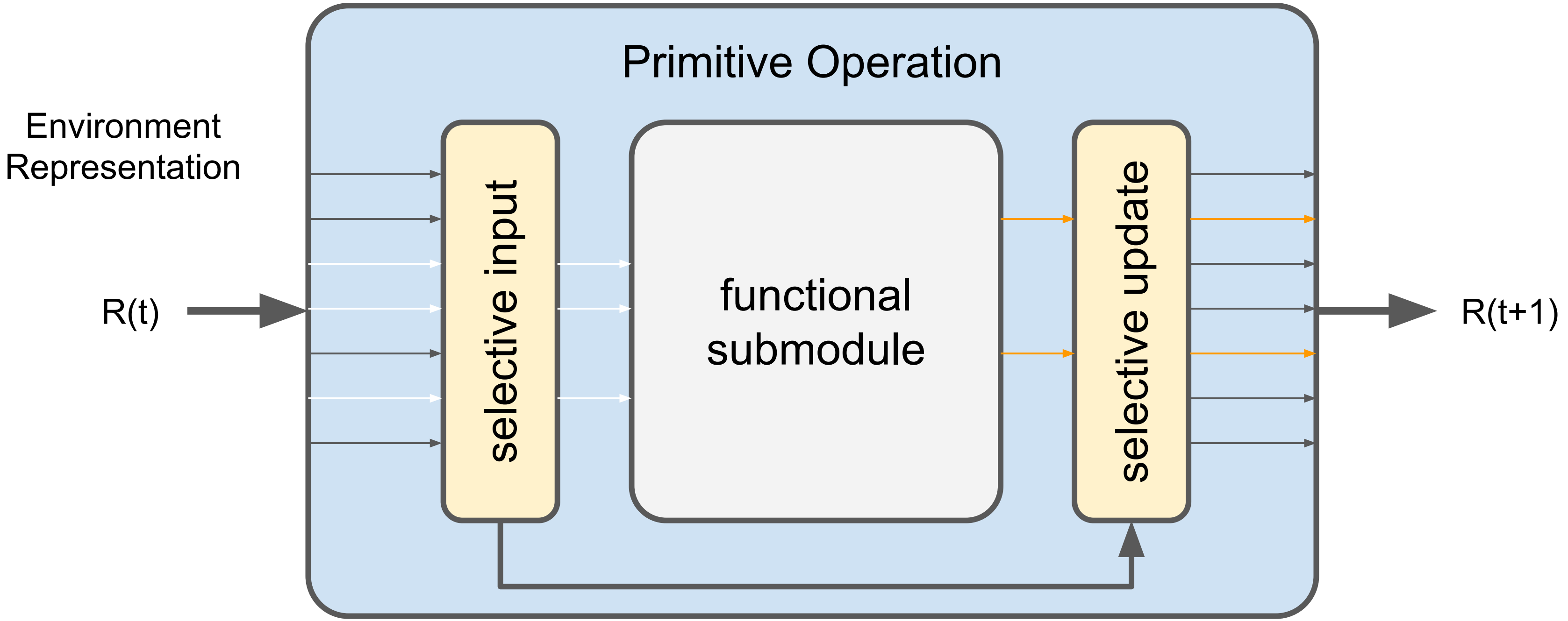
The Agent as a Problem Solver



Selection Sort: implementation

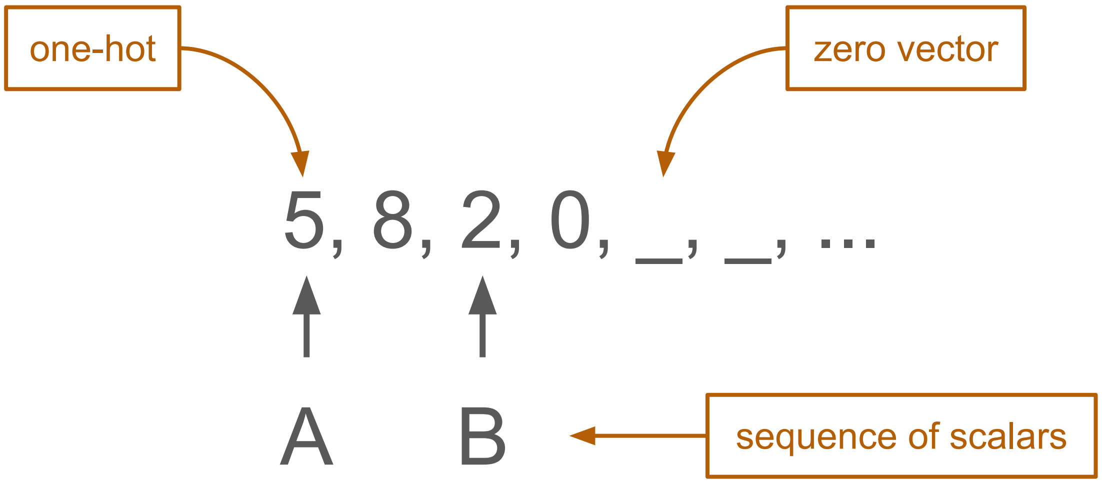


Decoupling of Functionality



Case Study
Experiments

Environment Representation



Primitive operations

- Move A

  2, 8, 5, _
  A A B

- Move B

  2, 8, 5, _
  A B B

- Return B

  2, 8, 5, _
  A B B

- Swap

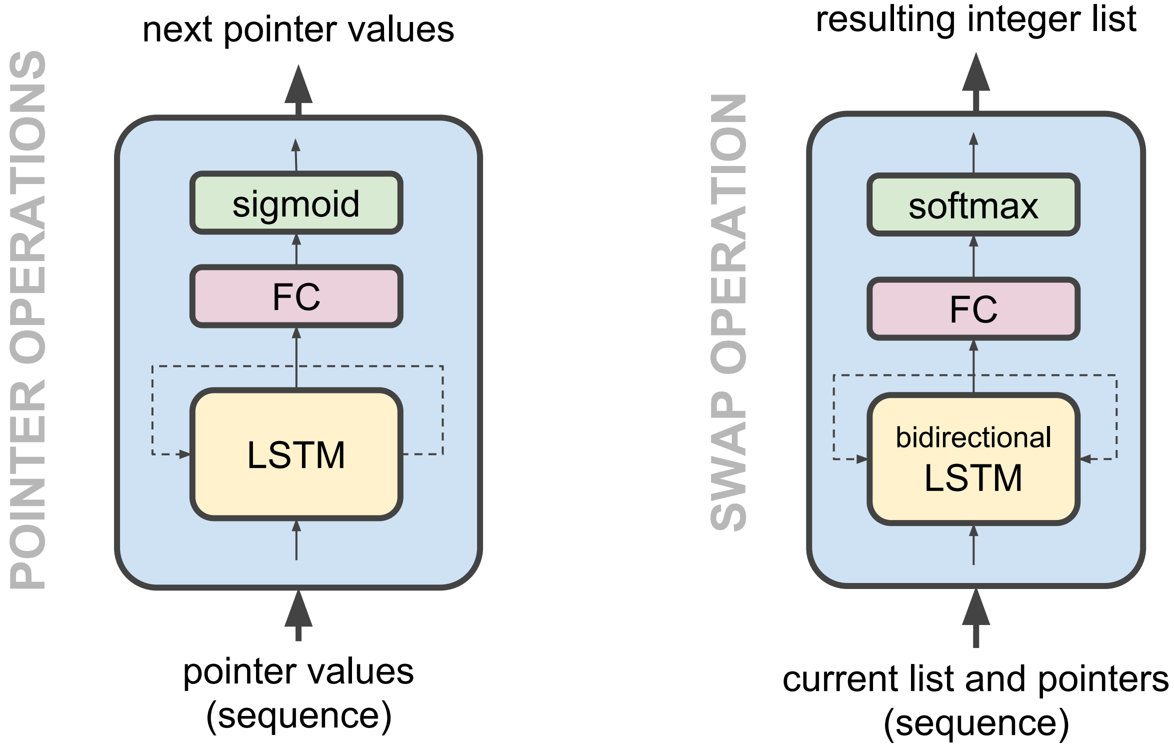
  5, 8, 2, _
  A B

- EOP

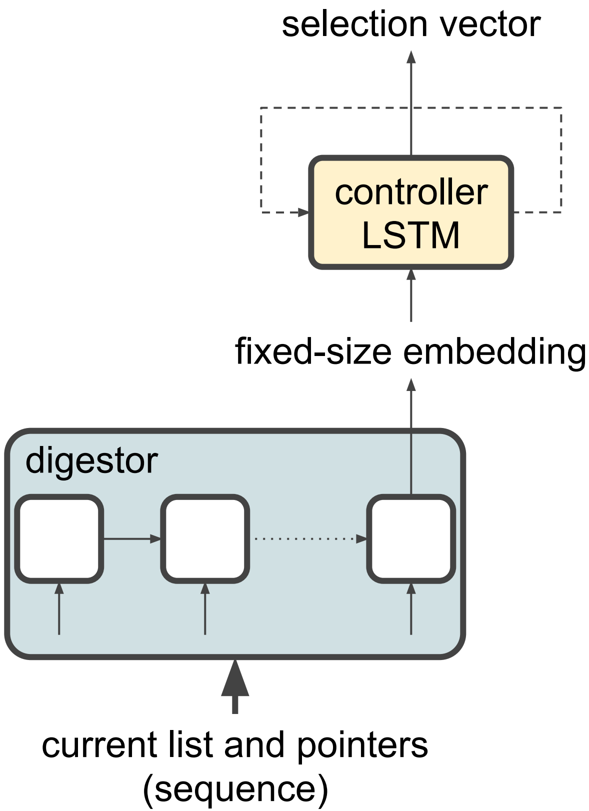
Tests

- Train & Assemble
- Monolithic training
- Time & data
- Generalization

Neural operations

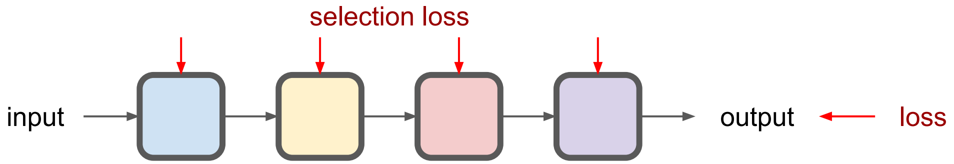


The control module

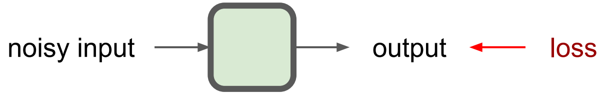


Training procedure

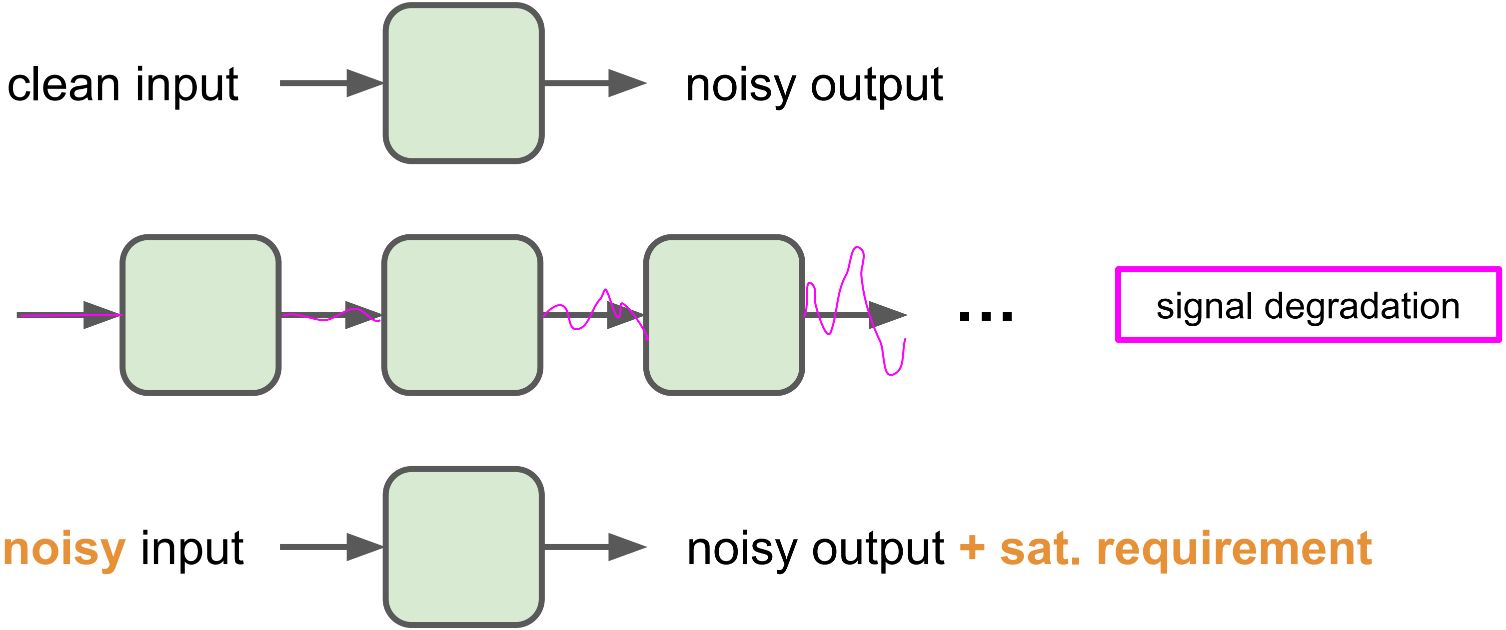
- Monolithic
  - Output loss
  - Selection loss
  - Stop criteria: mean error rate < 0.01 & loss slope



- Modular
  - Output loss
  - Input noise
  - Stop criteria: mean error rate < 0.01 & saturation error

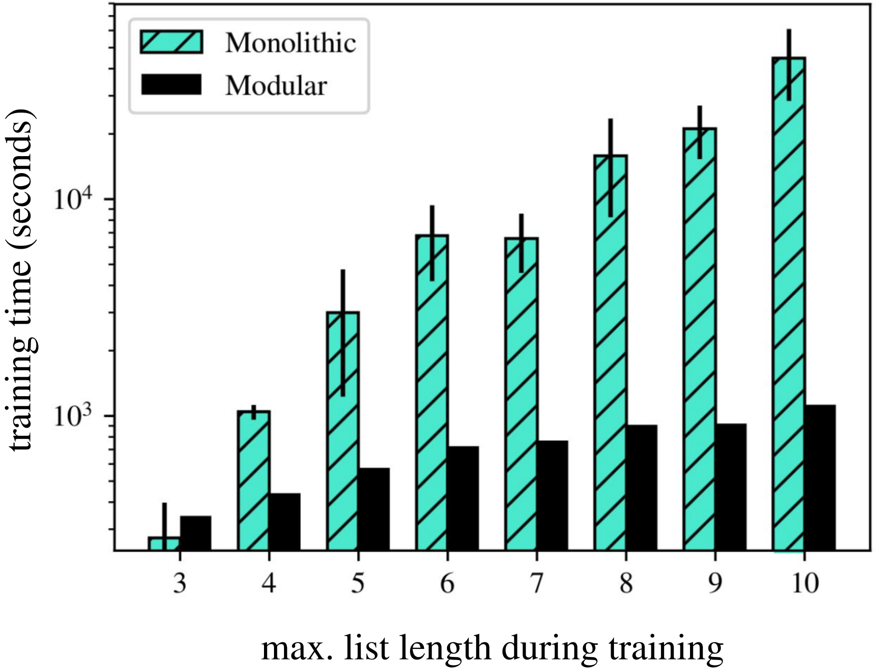


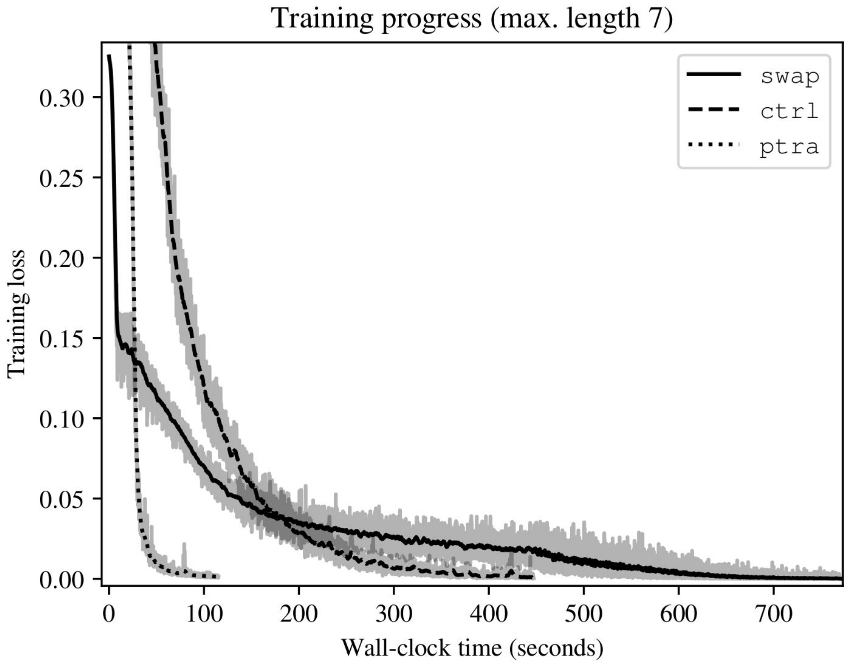
Module assembly



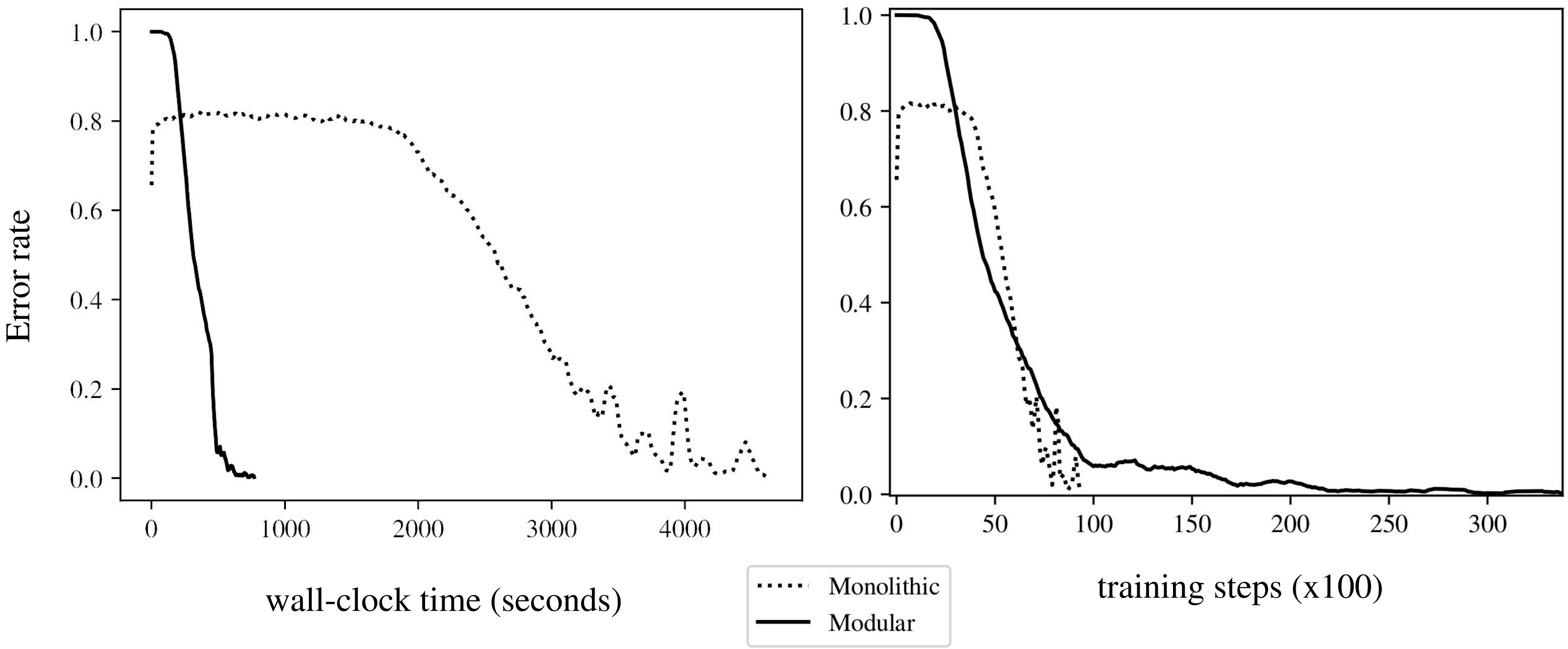
Results

Results: training time & smoothness

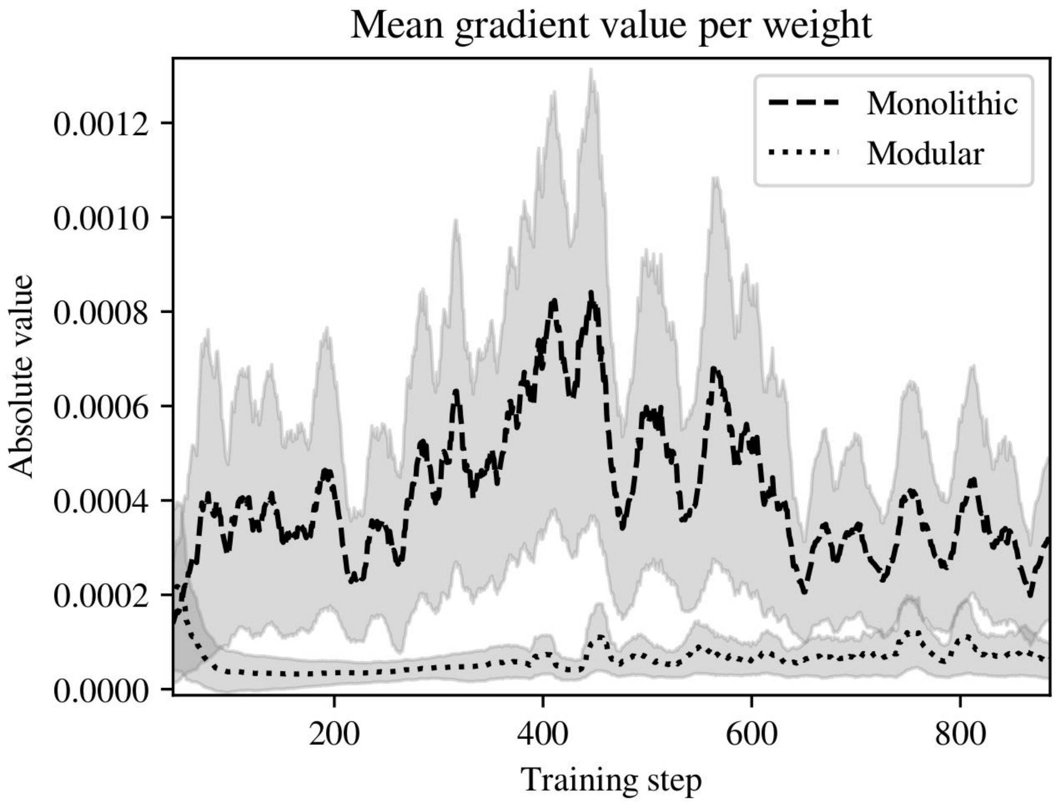




Results: step efficiency

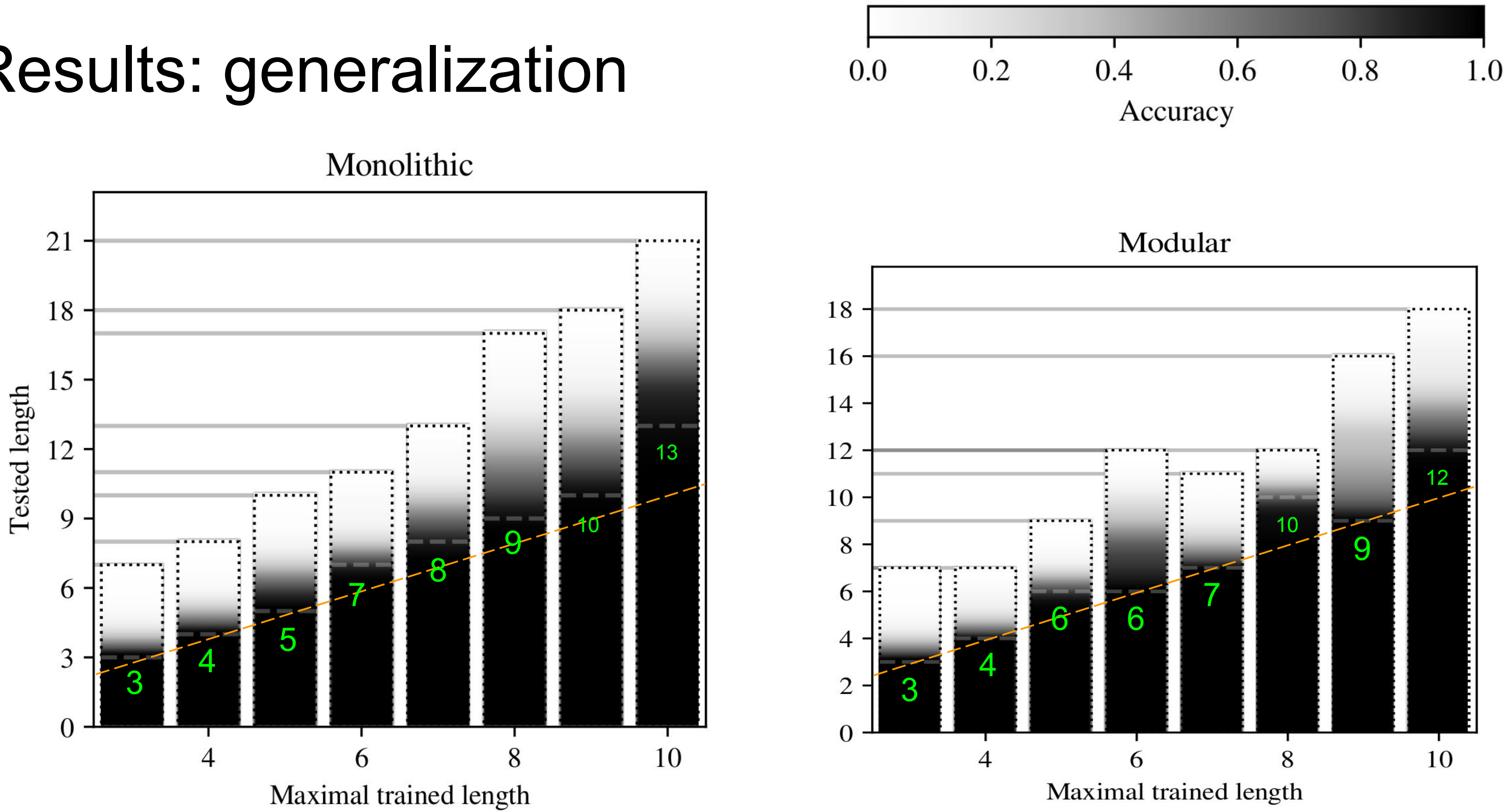


Results: gradient information



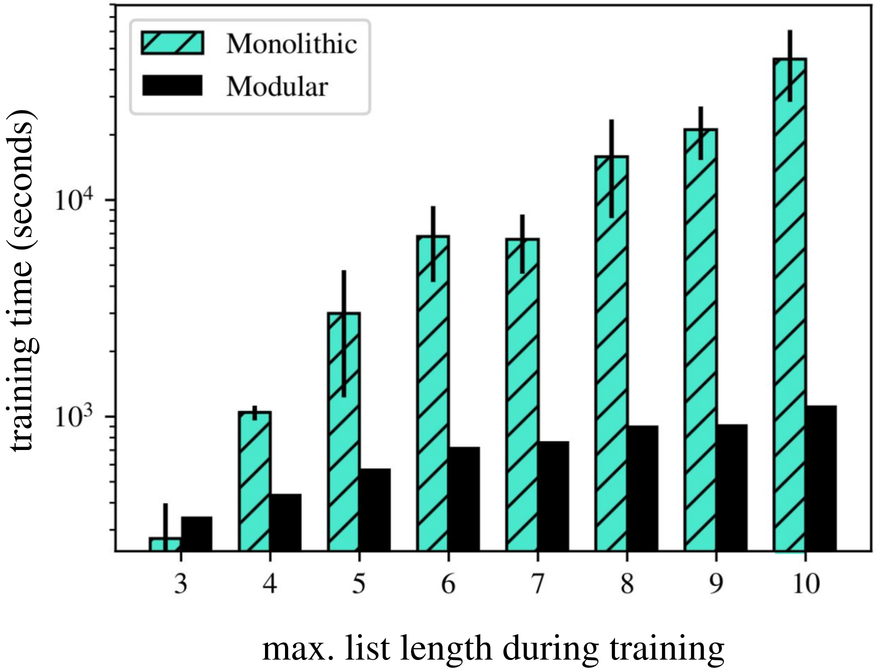
- Poorer gradient
- Higher learning rate
- Speed vs Quality

Results: generalization

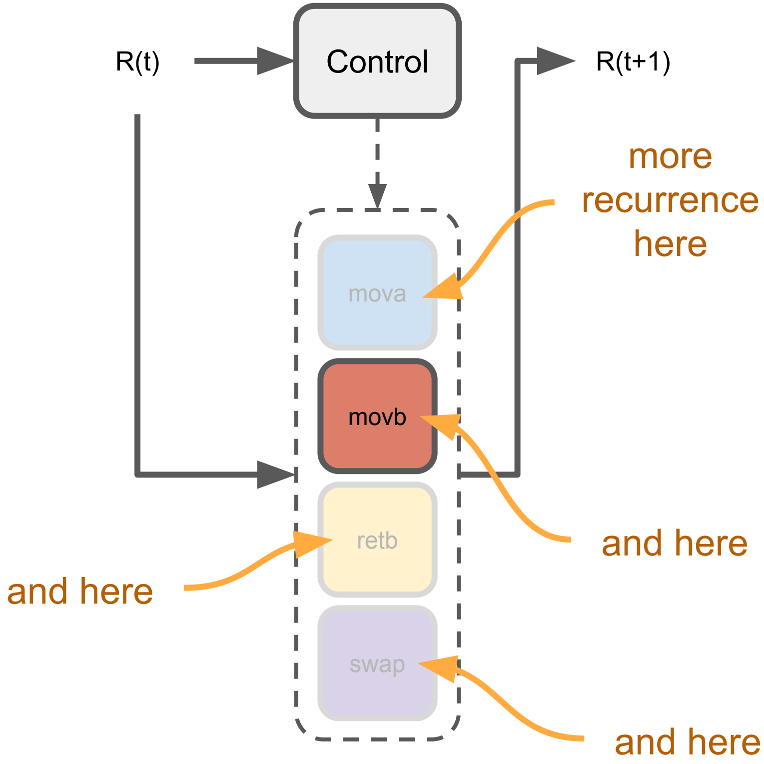


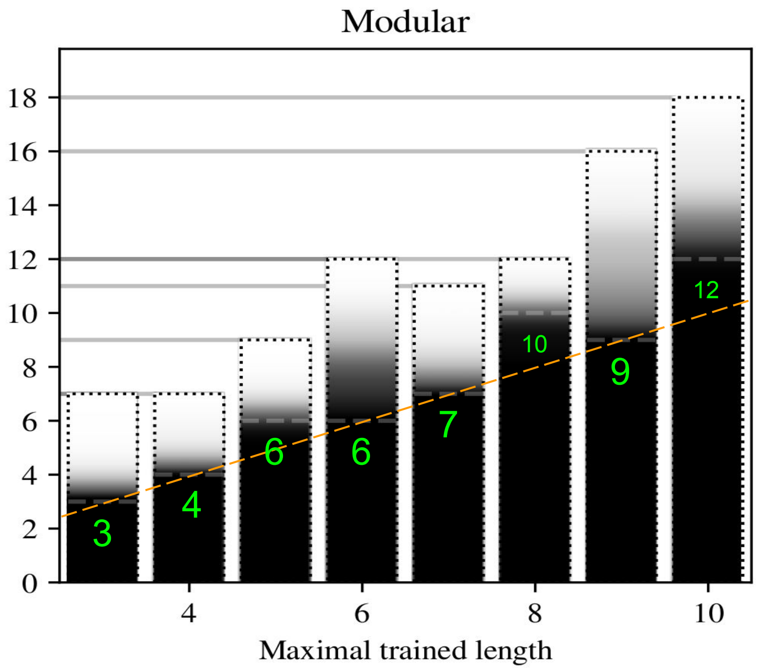
Conclusions

Conclusion 1: Time is precious

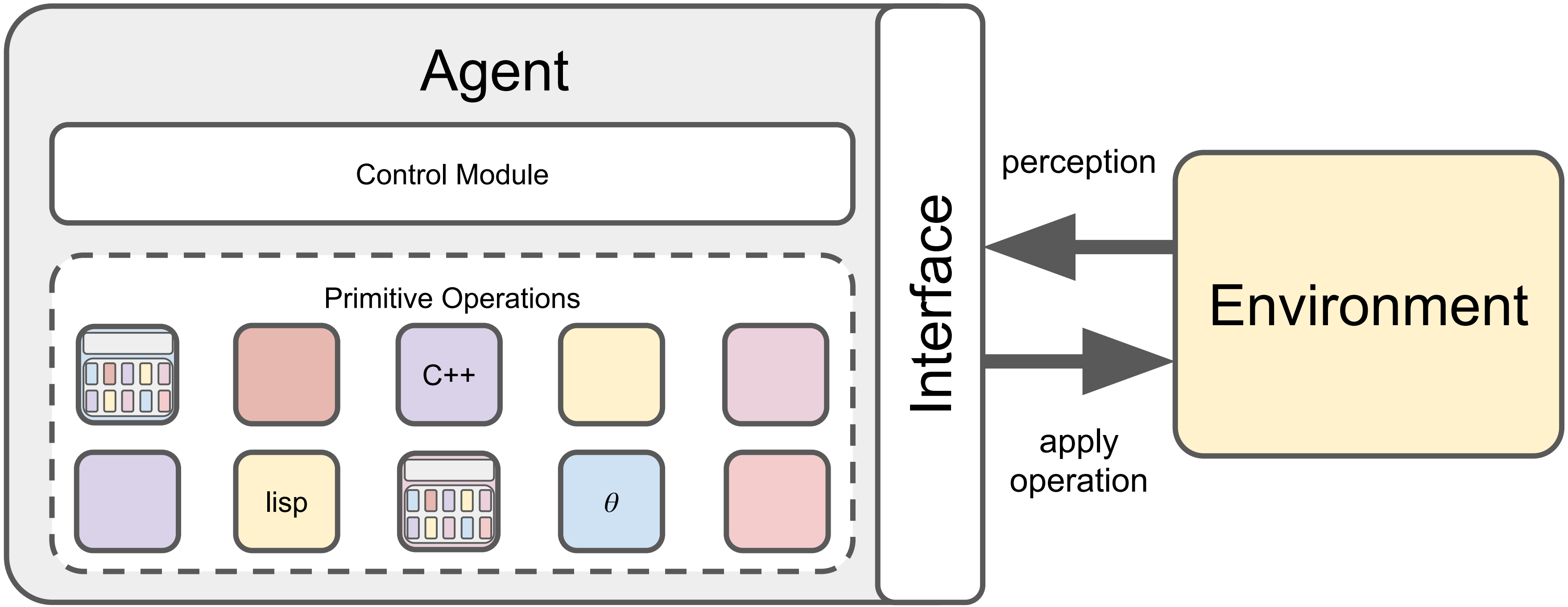


Conclusion 2: Reducing recurrence

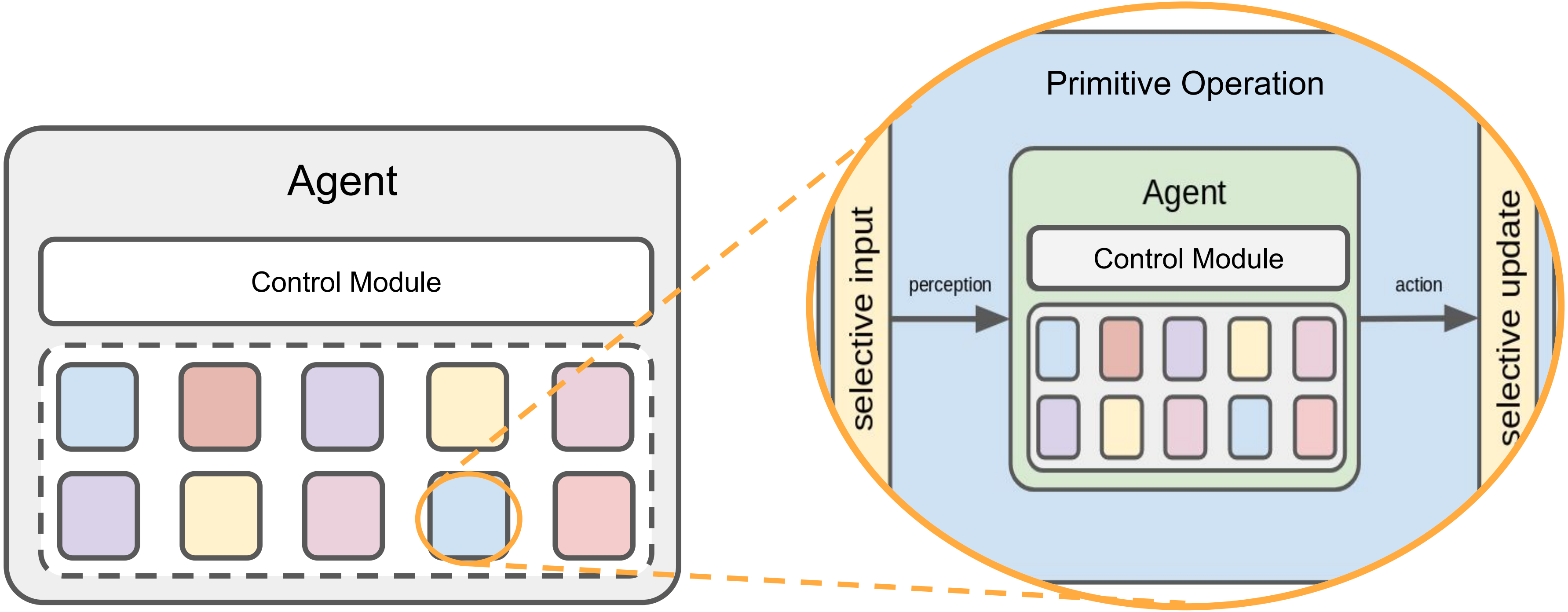




Conclusion 3: Complex Hybrid AI Systems

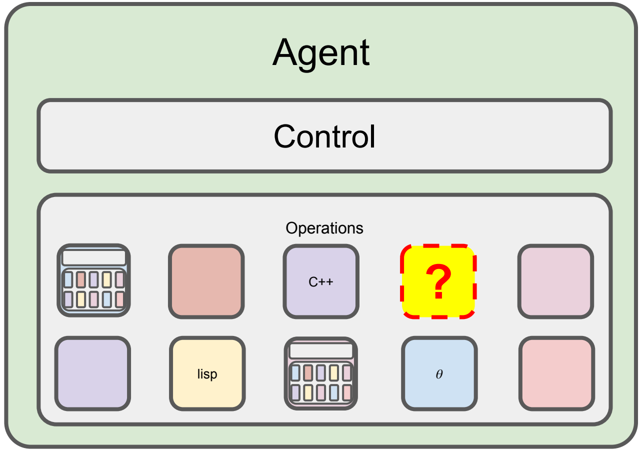


Conclusion 4: Hierarchical composition



Future research

- Transfer to state-of-the-art problems
- Reinforcement learning / Policy learning
- Reduce supervision
- Combine NNs with handcrafted modules
- Explore recursive compositionality
- Synthetic gradient for inserting blank modules



Modularity as a Means for Complexity Management in Neural Networks Learning

David Castillo-Bolado
Cayetano Guerra-Artal
Mario Hernández-Tejera

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

INSTITUTO UNIVERSITARIO SIANI INGENIERIA COMPUTACIONAL