A Simple Neural Attentive Meta-learning Reporting

K A N G   S U   H Y E O N

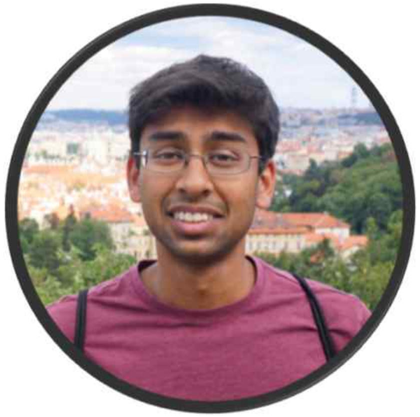
shqk023@gmail.com

Content

- **About paper**
- **Introduction**
- **Meta-learning preliminaries**
- **A Simple Neural Attentive Learner**
  - **Architecture**
  - **Modular building Blocks**
- **Experiments**
  - **Supervised learning**
  - **Reinforcement learning**
- **Conclusion and Future work**

About paper

UC Berkeley, Department of Electrical Engineering and Computer Science Embodied Intelligence



Nikhil Mishra
(OpenAI)

Mostafa Rohaninejad
(Embodied Intelligence)

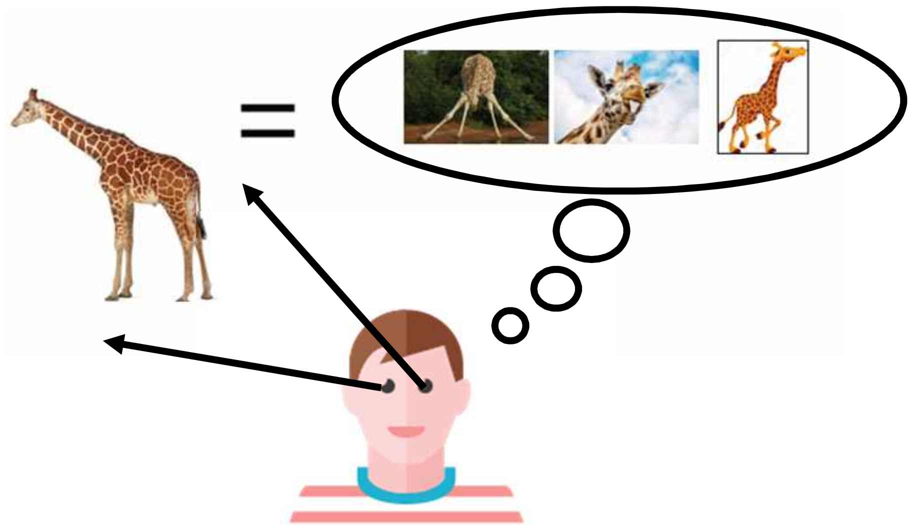
Xi Chen
(OpenAI)

Pieter Abbeel
(UC Berkeley)

Introduction

Human

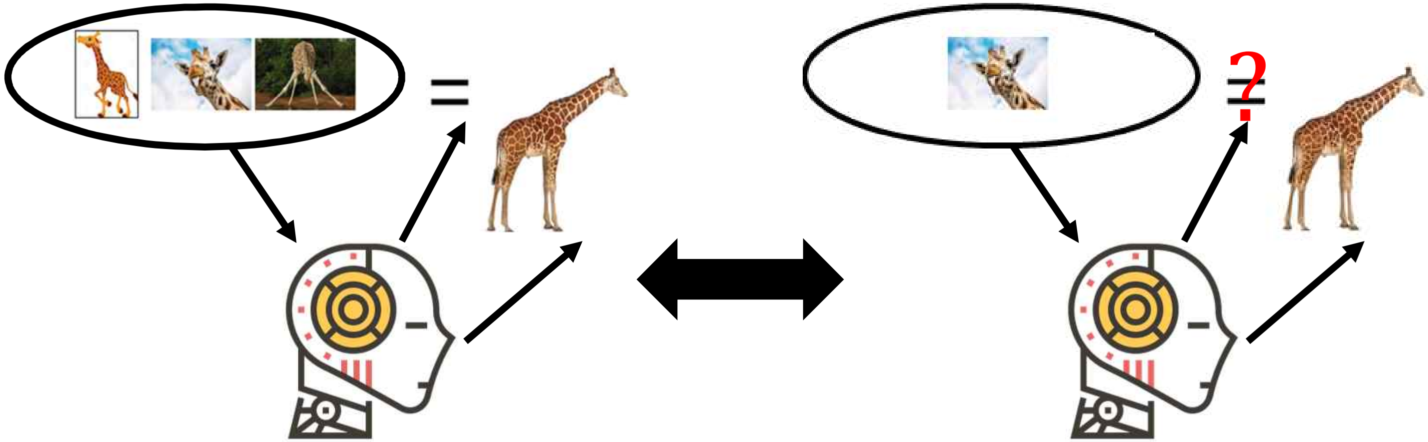
- **Humans effectively utilize prior knowledge and experiences to learn new skills quickly**



Introduction

Artificial Learner

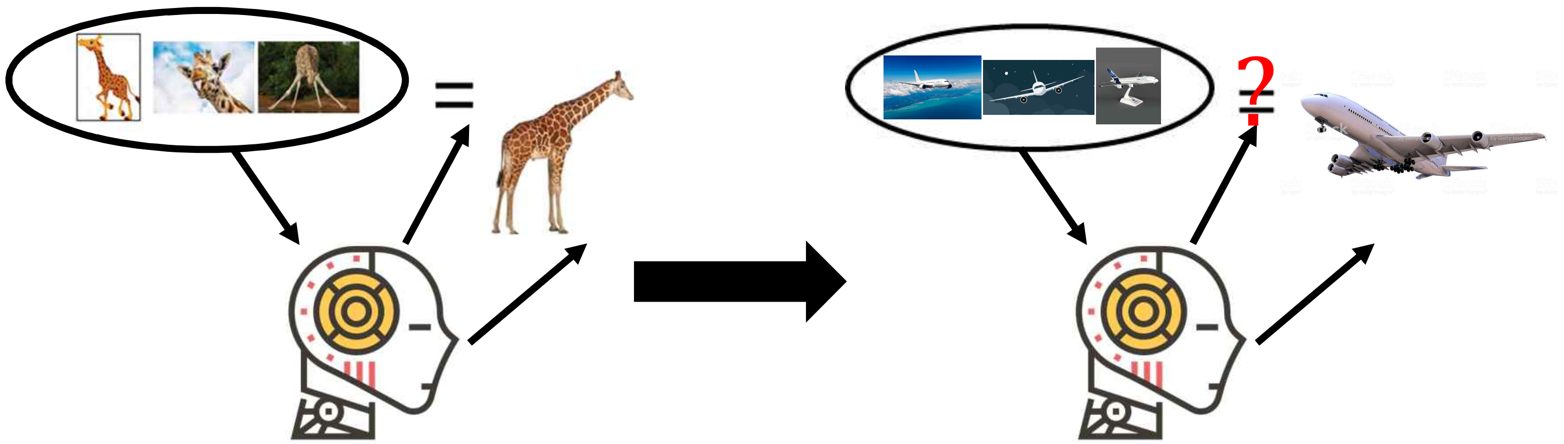
- **Artificial learners trained with traditional supervised or reinforcement learning methods generally perform poorly when only a small amount of data is available**



Introduction

Artificial Learner

- **Artificial learners trained with traditional supervised or reinforcement learning methods generally perform poorly when they need to adapt to a changing task**



Introduction

Meta Learning

- **Meta-learning seeks to resolve this deficiency by broadening the learner's scope to a distribution of related tasks.**
  - <u>Existing Learner</u>
    Training the learner on a single task
    (with the goal of generalizing to unseen samples from a similar data distribution)
  - <u>Meta-learner</u>
    Training the meta-learner on a distribution of similar tasks
    (with the goal of learning a strategy that generalizes to related but unseen tasks from a similar task distribution)

Introduction

Meta Learning

- The meta-learner should have the flexibility to learn the best way to solve the tasks it is presented with

- A meta-learner would need to have an expressive, versatile model architecture, in order to learn a range of strategies in a variety of domains

- Meta-learning can be formalized as a sequence to sequence problem; in existing approaches that adopt this view, the bottleneck is in the meta-learner's ability to internalize and refer to past experience

SNAIL

Propose a model architectures

Combine Temporal Convolution & Causal attention

Introduction

Meta Learning

- SNAIL(combine Temporal Convolution & Causal attention)

Enable the meta-learner to aggregate contextual information from past experience

Allow it to pinpoint specific pieces of information within that context

Evaluate

- Supervised Learning ➔ Omniglot, Mini-ImageNet
- Reinforcement Learning ➔ Multi-armed bandits, Tabular MDP,
  Visual navigation,
  Continuous control

Meta-learning Preliminaries

Notation & Formalization

T_i : $Task$ $(or\ Episode)$ $\sim \mathrm{T} = P(\mathrm{T}_i)$
x_t : $input$ $\sim P_i(x_t | x_{t-1}, a_{t-1})$
a_t : $output$ $\sim \pi(a_t | x_1, \dots, x_t; \theta)$
$\mathcal{L}_i(x_t, a_t)$: $Loss\ function$
H_i : $Episode\ Length$

$Meta-learner's\ objective$

$$\min_{\theta} E_{\mathrm{T}_i \sim \mathrm{T}} \left[\sum_{t=0}^{H_i} \mathcal{L}_i(x_t, a_t) \right]$$

Training → **A Meta-learner is trained by optimizing this expected loss over tasks sampled from T**

Testing → **The Meta-learner is evaluated on unseen tasks from a different task distribution $\widetilde{\mathrm{T}} = P(\widetilde{\mathrm{T}}_i)$ that is similar to the training task distribution T**

A Simple Neural Attentive Learner

SNAIL

- **Key Principle Motivation ➔ Simplicity & Versatility**

- **It should be generic and expressive enough to learn an optimal strategy, rather than having the strategy already built-in**

- **Several approach to the Meta-runner**
  - ➢ **Santoro et al : Using RNN**
    - But ➔ **Temporally-linear dependency bottleneck**
  - ➢ **Van den Oord et al : Using Temporal Convolution(dilated 1-D Conv)**
    - But ➔ **They have coarser access to inputs that are further back in time**
  - ➢ **Vaswani et al : Using Soft attention**
    - But ➔ **The lack of positional dependence**

A Simple Neural Attentive Learner

SNAIL

- SNAIL(combine $\boxed{\text{Temporal Convolution}}$ & $\boxed{\text{Causal attention}}$)
  - ➢ TC and Attention complement each other

Provide high-bandwidth access at the expense of finite context size

Provide pinpoint access over an infinitely large context

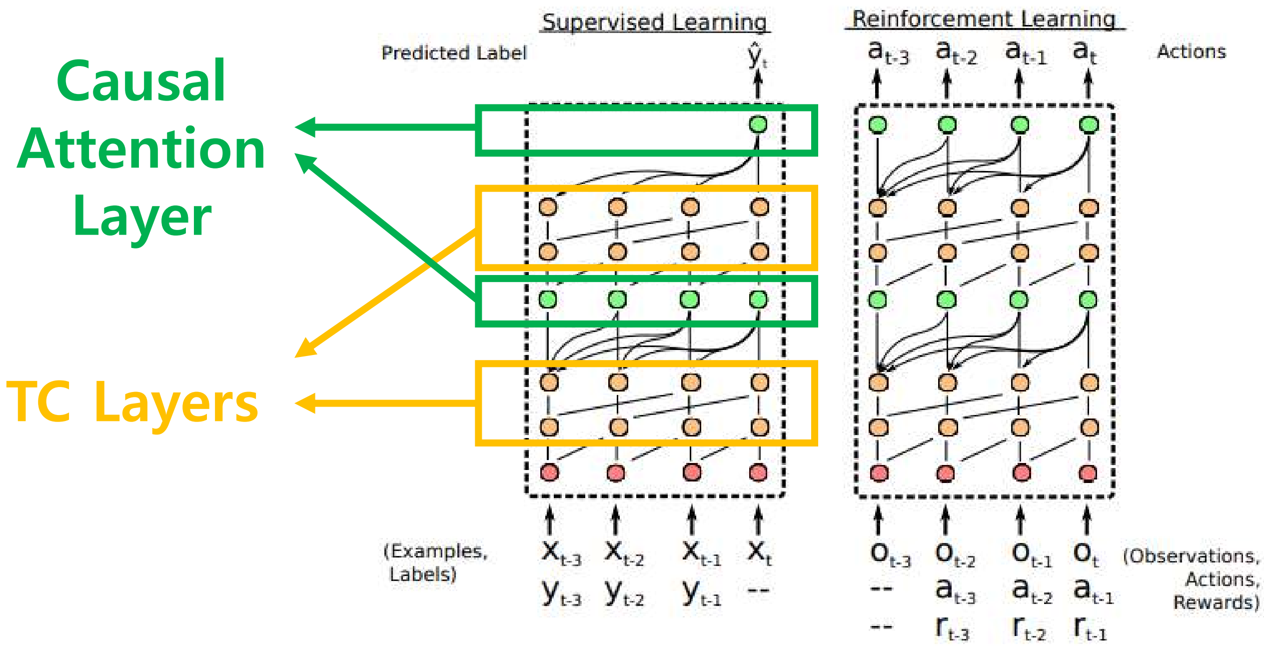
  - ➢ SNAIL is easier to train than traditional RNNs such as LSTM or GRUs

  - ➢ SNAIL can be efficiently implemented so that an entire sequence can be processed in a single forward pass

A Simple Neural Attentive Learner

SNAIL

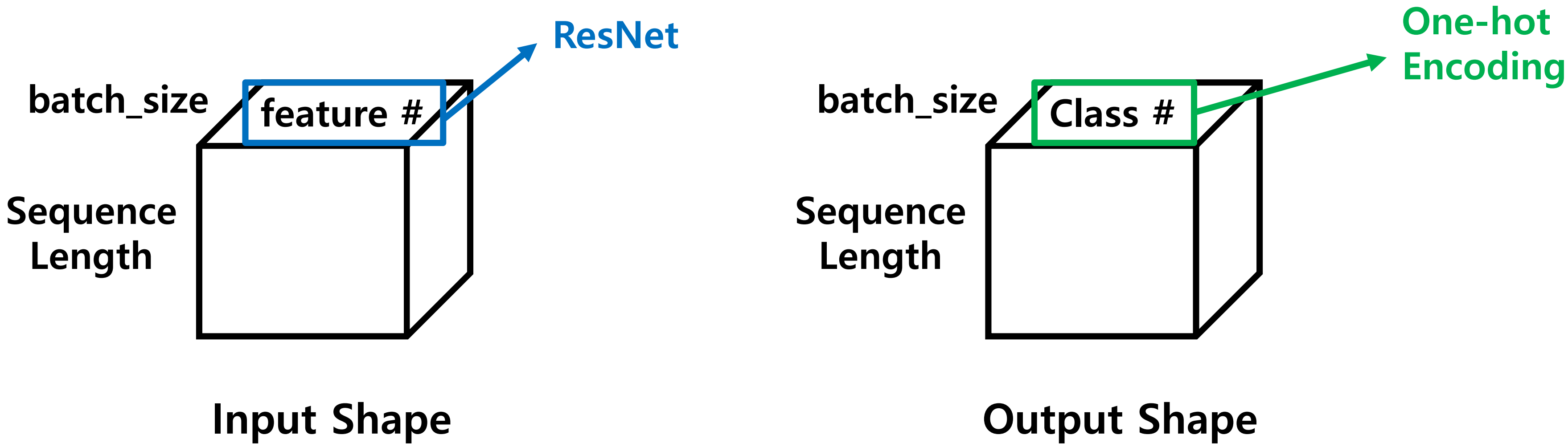
- **SNAIL Architecture**



A Simple Neural Attentive Learner

Modular Building Blocks

- **Data Input Shape & Output Shape**



A Simple Neural Attentive Learner

Modular Building Blocks

- **Dense Block**

function DENSEBLOCK(inputs, dilation rate R, number of filters D):
    xf, xg = CausalConv(inputs, R, D), CausalConv(inputs, R, D)
    activations = tanh(xf) * sigmoid(xg)
    return concat(inputs, activations)

- **TCBlock**

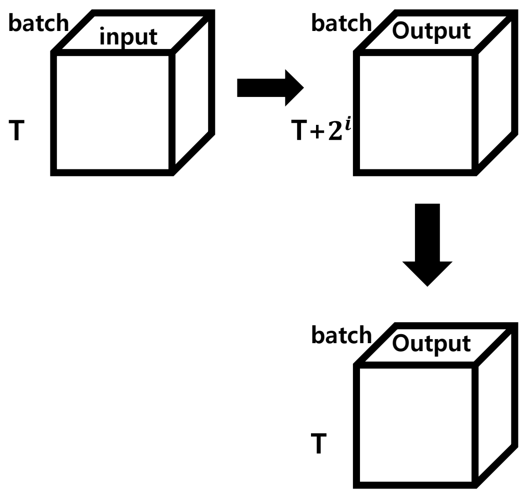
function TCBLOCK(inputs, sequence length T, number of filters D):
    for i in $1, \ldots, \lceil \log_2 T \rceil$ **do**
        inputs = DenseBlock(inputs, 2^i, D)
    return inputs

A Simple Neural Attentive Learner

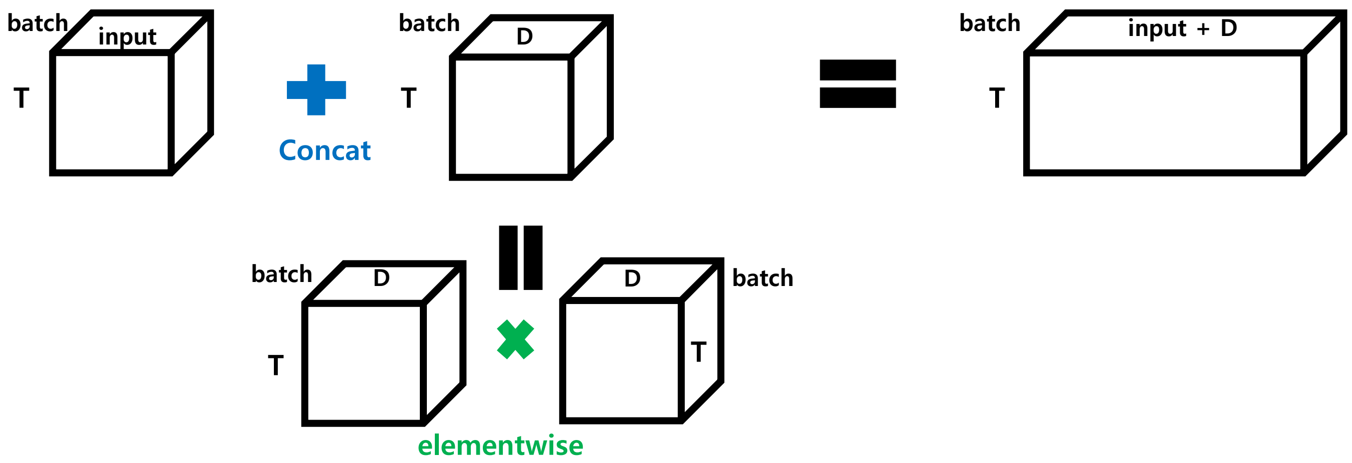
Modular Building Blocks

- **Dense Block & TCBlock Process**

<Conv 1-D>



<Dense Block>

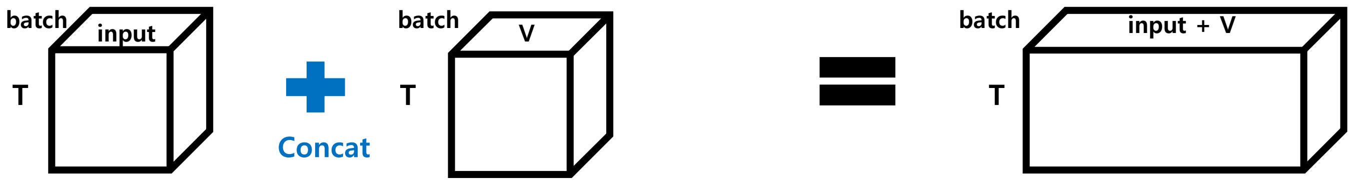


A Simple Neural Attentive Learner

Modular Building Blocks

- **Attention Block**

```
function AttentionBlock(inputs, key size K, value size V):
    keys, query = affine(inputs, K), affine(inputs, K)
    logits = matmul(query, transpose(keys))
    probs = CausallyMaskedSoftmax(logits / √K)
    values = affine(inputs, V)
    read = matmul(probs, values)
    return concat(inputs, read)
```



Experiments

Experiment Purpose

- **How does SNAIL's generality affect its performance on a range of meta-learning tasks?**
- **How does its performance compare to existing approaches that are specialized to a particular task domain, or have elements of a high-level already built-in?**
- **How does SNAIL scale with high-dimensional inputs and long-term temporal dependencies?**

Experiments

Few-Shot Image Classification

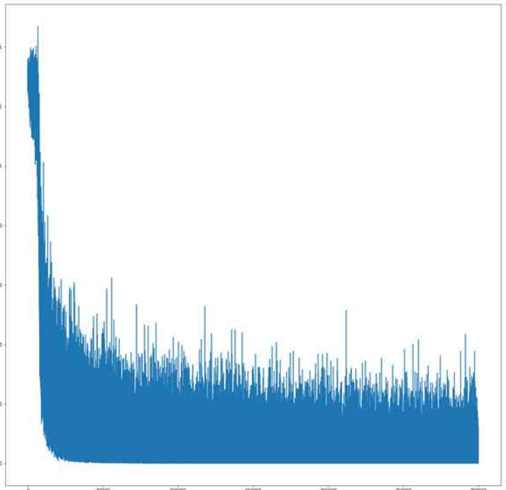
- **Omniglot Dataset**

Experiments

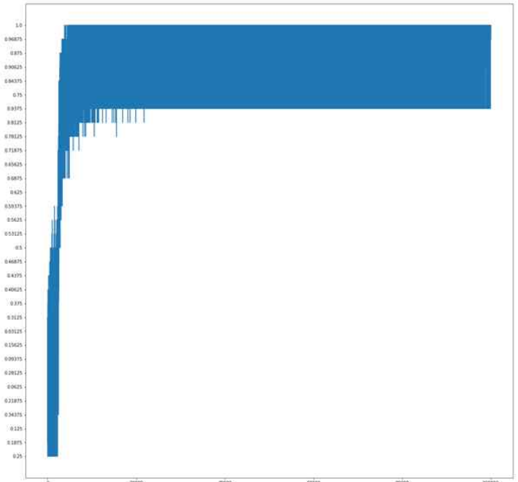
Few-Shot Image Classification

- **Omniglot Dataset(epoch:100 / iteration:10000)**

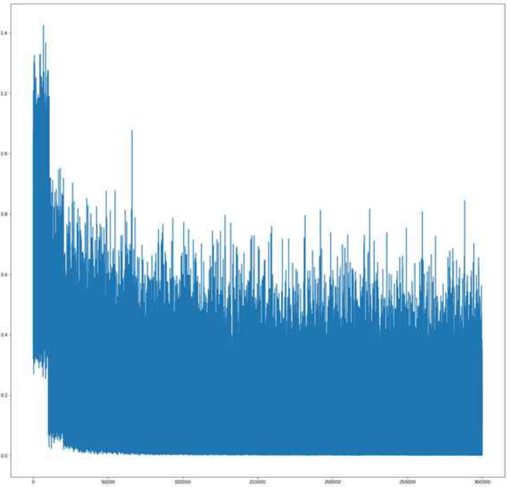
Train loss



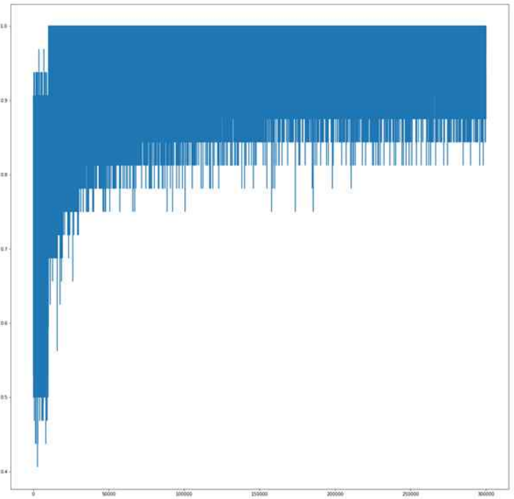
Train acc



Val loss



Val acc



Experiments

Few-Shot Image Classification

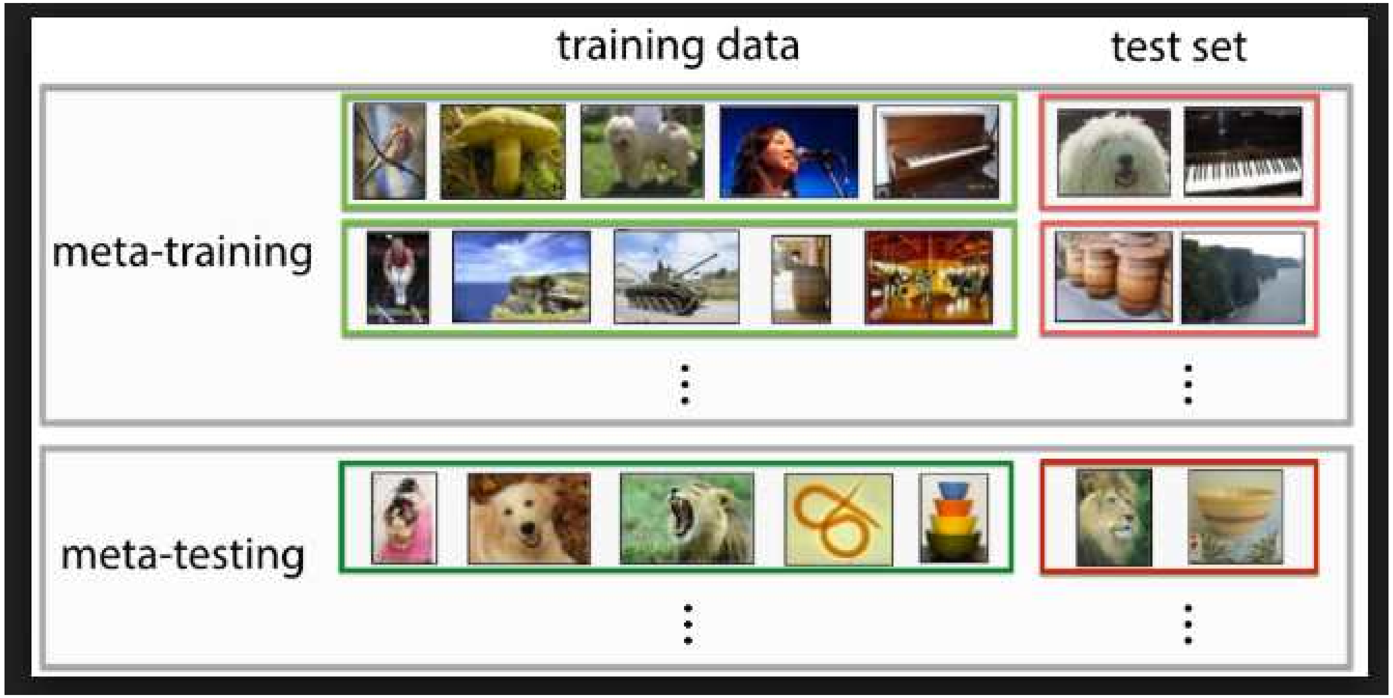
- **Omniglot Dataset**

Model	1-shot 5-way (Acc)	5-shot 5-way (Acc)	1-shot 20-way (Acc)	1-shot 20-way (Acc)
Paper	99.07%	99.78%	97.64%	99.36%
Repo	98.31%	99.26%	93.75%	97.88%

Experiments

Few-Shot Image Classification

- **Mini-ImageNet Dataset**



Experiments

Few-Shot Image Classification

- **Mini-ImageNet Dataset**

Method	5-Way Mini-ImageNet	
	1-shot	5-shot
Vinyals et al. (2016)	43.6%	55.3%
Finn et al. (2017)	48.7% $\pm$ 1.84%	63.1% $\pm$ 0.92%
Ravi & Larochelle (2017)	43.4% $\pm$ 0.77%	60.2% $\pm$ 0.71%
Snell et al. (2017)	46.61% $\pm$ 0.78%	65.77% $\pm$ 0.70%
Munkhdalai & Yu (2017)	49.21% $\pm$ 0.96%	–
SNAIL, Ours	**55.71% $\pm$ 0.99%**	**68.88% $\pm$ 0.92%**

Experiments

Reinforcement Learning

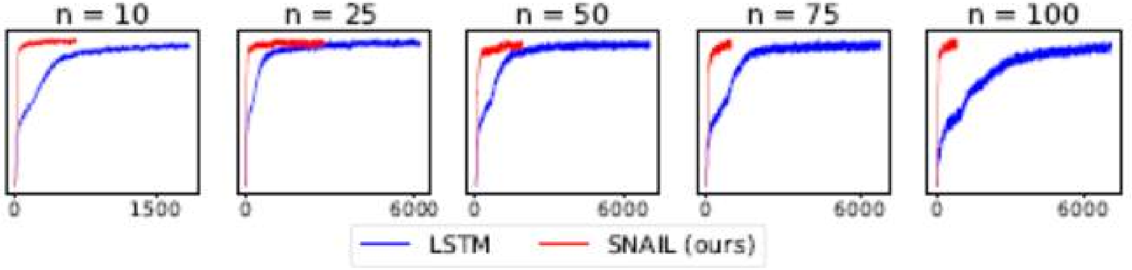
- **Multi-armed Bandits**

Setup (N, K)	Method Gittins (optimal as $N \to \infty$)	Random	LSTM	MAML	SNAIL (ours)
$10, 5$	**6.6**	5.0	**6.7**	6.5 ± 0.1	$\mathbf{6.6 \pm 0.1}$
$10, 10$	**6.6**	5.0	**6.7**	$\mathbf{6.6 \pm 0.1}$	$\mathbf{6.7 \pm 0.1}$
$10, 50$	6.5	5.1	**6.8**	$\mathbf{6.6 \pm 0.1}$	$\mathbf{6.7 \pm 0.1}$
$100, 5$	**78.3**	49.9	**78.7**	67.1 ± 1.1	$\mathbf{79.1 \pm 1.0}$
$100, 10$	**82.8**	49.9	**83.5**	70.1 ± 0.6	$\mathbf{83.5 \pm 0.8}$
$100, 50$	**85.2**	49.8	**84.9**	70.3 ± 0.4	$\mathbf{85.1 \pm 0.6}$
$500, 5$	**405.8**	249.8	401.5	–	$\mathbf{408.1 \pm 4.9}$
$500, 10$	**437.8**	249.0	**432.5**	–	$\mathbf{432.4 \pm 3.5}$
$500, 50$	**463.7**	249.6	438.9	–	442.6 ± 2.5
$1000, 50$	**944.1**	499.8	847.43	–	889.8 ± 5.6

Experiments

Reinforcement Learning

- **Tabular MDPs**

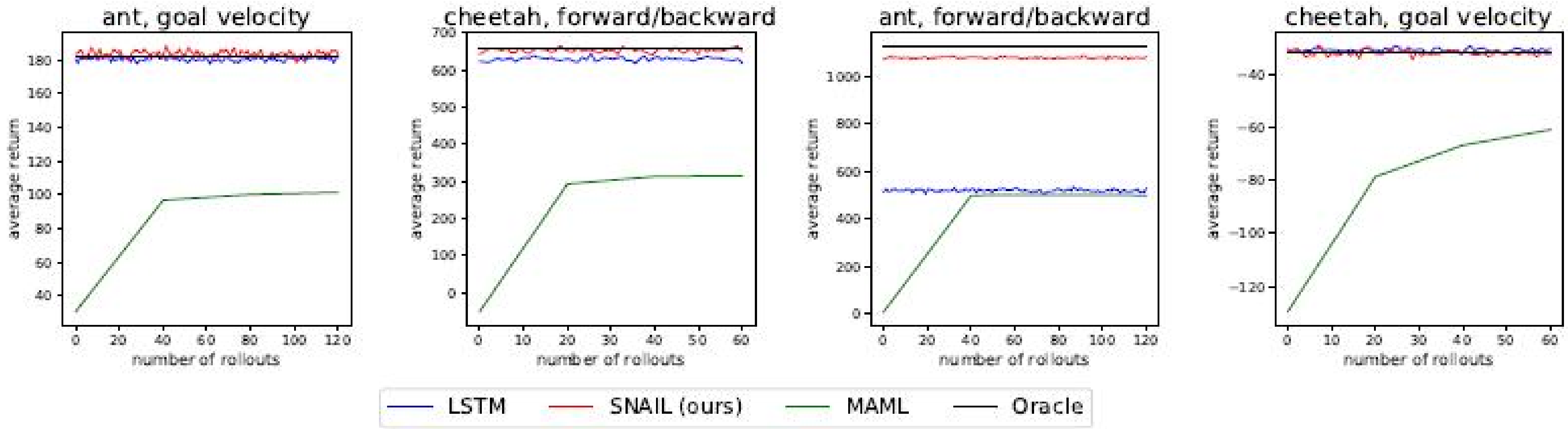


N	Method							
	Random	ϵ-greedy	PSRL	OPSRL	UCRL2	LSTM	MAML	SNAIL (ours)
10	0.482	0.640	0.665	0.694	0.706	0.752	0.563	**0.766 ± 0.001**
25	0.482	0.727	0.788	0.819	0.817	0.859	0.591	**0.862 ± 0.001**
50	0.481	0.793	0.871	0.897	0.885	0.902	–	**0.908 ± 0.003**
75	0.482	0.831	0.910	**0.931**	0.917	0.918	–	**0.930 ± 0.002**
100	0.481	0.857	0.934	**0.951**	0.936	0.922	–	0.941 ± 0.003

Experiments

Reinforcement Learning

- **Continuous Control**

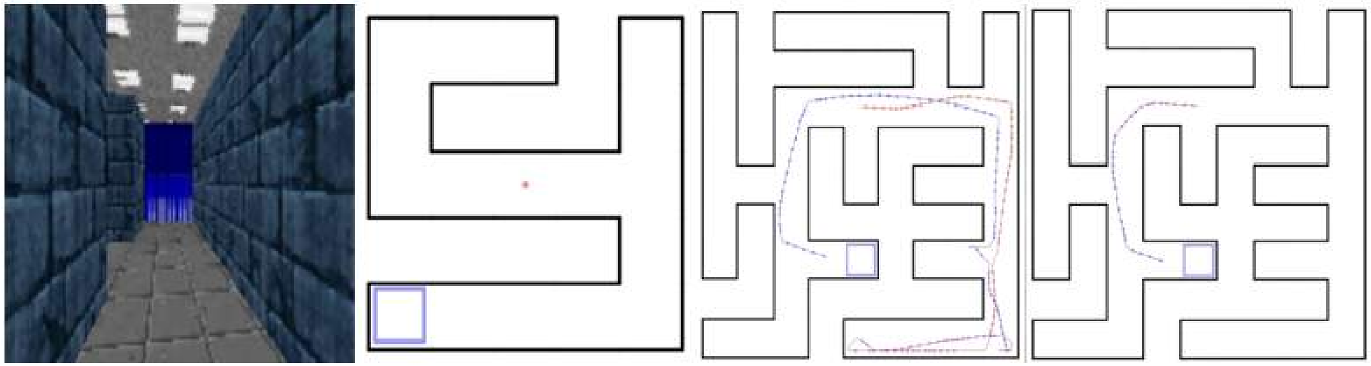


Experiments

Reinforcement Learning

- **Visual Navigation**

Method	Small Maze Episode 1	Small Maze Episode 2	Large Maze Episode 1	Large Maze Episode 2
Random	188.6 ± 3.5	187.7 ± 3.5	420.2 ± 1.2	420.8 ± 1.2
LSTM	52.4 ± 1.3	39.1 ± 0.9	180.1 ± 6.0	150.6 ± 5.9
SNAIL (ours)	$\mathbf{50.3 \pm 0.3}$	$\mathbf{34.8 \pm 0.2}$	$\mathbf{140.5 \pm 4.2}$	$\mathbf{105.9 \pm 2.4}$



Conclusion and Future Work

Conclusion

- We presented a simple and generic class of architectures for meta-learning, motivated by the need for a meta-learner to quickly incorporate and refer to past experience.
- Our simple neural attentive learner (SNAIL) utilizes a novel combination of temporal convolutions and causal attention, two building blocks of sequence-to-sequence models that have complementary strengths and weaknesses

Conclusion and Future Work

Conclusion

- **Another interesting idea would be to train an meta-learner that can attend over its entire lifetime of experience (rather than only a few recent episodes, as in this work)**
- **An agent with this lifelong memory could learn faster and generalize better; however, to keep the computational requirements practical, it would also need to learn how to decide what experiences are worth remembering.**

Conclusion and Future Work

Future Work

- Although we designed SNAIL with meta-learning in mind, it would likely excel at other sequence-to-sequence tasks, such as language modeling or translation; we plan to explore this in future work

THANK YOU

Q & A