Meta Attention Networks: Meta Learning Attention To Modulate Information Between Sparsely Interacting Recurrent Modules

Kanika Madan [1] **Rosemary Nan Ke** [1 2] **Anirudh Goyal** [1] **Yoshua Bengio** [1 3]

Abstract

Decomposing knowledge into interchangeable pieces promises a generalization advantage when, at some level of representation, the learner is likely to be faced with situations requiring novel combinations of existing pieces of knowledge or computation. We hypothesize that such a decomposition of knowledge is particularly relevant for higher levels of representation as we see this at work in human cognition and natural language in the form of systematicity or systematic generalization. To study these ideas, we propose a particular training framework in which we assume that the pieces of knowledge an agent needs, as well as its reward function are stationary and can be re-used across tasks and changes in distribution. As the learner is confronted with variations in experiences, the attention selects which modules should be adapted and the parameters of those *selected* modules are adapted fast, while the parameters of attention mechanisms are updated slowly as meta-parameters. We find that both the meta-learning and the modular aspects of the proposed system greatly help achieve faster learning in experiments with reinforcement learning setup involving navigation in a partially observed grid world.

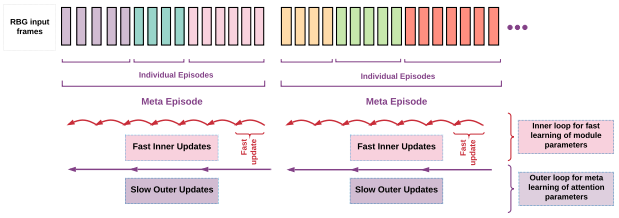

Figure 1. **Meta-Attention setup:** The fast and slow meta learning loops learn different parameters of the model at different timescales

1. Introduction

The classical framework for machine learning is focused on the framework of i.i.d. (identical and independent distributed data), meaning the test data has the same distribution as the training distribution. However, a learning agent that is interacting with the world is always facing non-stationarities because of the actions of the agent itself, or because of the other agents. Having a model that can better handle such changes and generalize better has been a long standing goal of machine learning. At the same time, most of the current deep learning systems are built in the form of one big network, consisting of a layered but otherwise monolithic structure, which could lead to co-adaptation across different components of the network, and thus requiring changes to most of these components as the task or the distribution changes, potentially leading to catastrophic interferences between different pieces of knowledge.

Humans, however, seem to be able to learn a new task quickly by re-using previous knowledge. This raises two fundamental questions which we explore here: (1) how to separate knowledge into easily recomposable pieces or modules and (2) how to do this so as to achieve fast adaptation to new tasks or changes in distribution when a module may need to be modified, or when modules may need to be combined in novel ways. In this paper, we study the systematic generalization of deep neural networks to tasks which are unseen. We show how the proposed agent can generalize better not only on the seen data, but also is more sample efficient, faster to train and adapt, and has better transfer capabilities to changes in distributions. We show strong evidence that combining meta-learning with modular architectures can help in building smarter agents, which not only understand their environment better, but can also learn and leverage the compositional properties of the system to generalize better on unseen domains and achieve better transferability and generalization in a more systematic manner.

[1]Mila, Université de Montréal [2]Polytechnique Montréal [3]CIFAR Senior Fellow. Correspondence to: Kanika Madan <madankanika.s@gmail.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

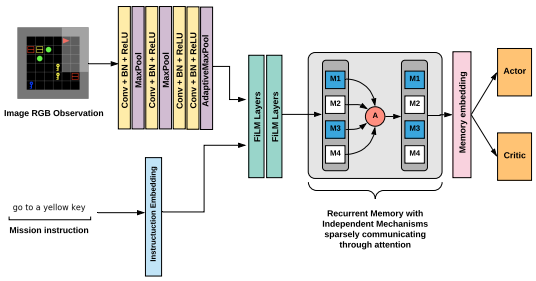

Figure 2. **Proposed Model Architecture:** Input images processed through an encoder and Mission instruction embedding passed through a set of independent recurrent modules which compete using attention to capture the dynamics of the system

2. Approach

The motivation behind this work is to investigate whether modular architectures, combined with learning different parts of the model at different timescales, can help in better learning which is not only more sample efficient, but also generalizes well across changes in task distributions. We find that this combination, in the proposed way, enables a more systematic generalization to new data regimes and a better transfer across changes in distributions.

The proposed framework is evaluated on grounded language learning tasks that involve training agents with language and visual input, with a focus on transfer to new tasks. The network consists of a modular neural network architecture that consists of an ensemble of recurrent components interacting with each other sparingly through a bottleneck of attention. To update the parameters of the proposed model, a meta-learning approach is used that updates different parts of the model at different timescales, see Fig. 1. The learning happens over a meta-episode in two phases as follows:

Fast learning phase: In order to quickly learn the dynamics of the environment, a subset of the modules, that are most relevant to the current input, are updated. The recurrent modules capture the underlying structure of the task distribution, and their parameters are updated multiple times by looking over several short interaction spans in the meta-episode.

Meta / Slow learning phase: The selection of modules that are relevant to the current input and the modulation of information exchange between these modules are performed using two types of attention mechanisms, an input attention and a communication attention. These attention mechanisms define which modules to activate, and how to combine them to enable a sparse communication and an appropriate information exchange. The parameters of these attention mechanisms are meta-learned, by using much longer spans of agent's experiences collected over the meta-episode to capture long-term dependencies and connectivity patterns of the modules. During this phase, the parameters of the recurrent modules are not changed. Since this phase looks at much longer time horizons, the number of updates to the attention parameters is lower, and the updates happen much more slowly.

We describe the different components of the model and the two learning phases in more details below. We hypothesize, and validate experimentally, that this approach of having modular networks in which different parts of the model are learnt over different timescales performs better in several aspects than a single large monolithic network in which all parameter updates happen at the same time.

2.1. Ensemble of Sparsely Interacting Modules

Ensemble of Interacting Modules: We follow the similar setup as RIMs (Goyal et al., 2019), which consists of an ensemble of modules, each operating with their own independent dynamics and interacting with each other through the bottleneck of attention. The proposed framework consists of a single layered recurrent structure such that at timestep t, the hidden state $\mathbf{h}_t$ is decomposed into n modules with their own independent hidden states $\mathbf{h}_{t,k}$ for $k = 1, ...n$ modules. Out of all these n modules, at any given timestep, only a subset of these modules are activated, and the updates for the hidden states follow a three-step process. First, a subset of modules is selectively activated based on their relevance to the current input. Second, the activated modules independently process the information made available to them using their internal dynamics. Third, the active modules communicate with the other modules through a bottleneck of attention to sparsely share information.

Selective Activation of Modules Using Input Attention: Out of the n modules, only a sparse subset k of them are active at any given point, and only the parameters of this subset of modules are updated in every update of the fast loop. Each module generates queries which are combined with the keys and values obtained from the concatenation of the actual input $\mathbf{x}_t$ and a dummy null input $\mathbf{\emptyset}$ to get attention scores and an attention modulated input. Based on these attention scores, a fixed number (k) of the n modules are activated. The modules that pay the least attention to a dummy null vector $\mathbf{\emptyset}$ in the input, i.e., pay the most attention to the actual input x_t, get activated.

Let $h_{t,j}$ represent the hidden state of the j^{th} module at timestep t, and θ_j represent the parameters of module j (different modules have different parameters). First, given an input, each module creates queries which are combined with the keys and values obtained from the input $\mathbf{x}_t$ to get an attention score for each module. These attention scores are then used to create a sparse subset of top k most relevant modules which get activated. Let this set of activated modules at timestep t be S_t, and let the updated hidden

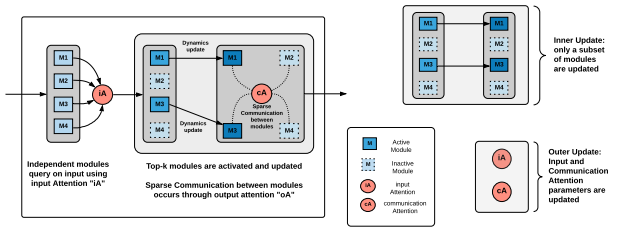

Figure 3. **Parameter updates in Fast and Slow loops**: The dynamics of the system are captured by a set of independent modules that sparsely communicate and compete with each other through the bottleneck of attention. In every fast update, only top k most relevant modules get updated. The two sets of attention parameters (input and communication attention) are updated in the slow loop

states of module j at time t be $\tilde{h}_{t,j}$. Then, each module in active set S_t updates its hidden state as per its *default* recurrent dynamics, D_k such that $\tilde{h}_{t,j} = D_k(h_{t,j})$, and the modules which are not activated have their hidden states remain unchanged as $\tilde{h}_{t,j} = h_{t,j}$.

Communication between different modules: Each of the activated module gets to interact with all other modules, each of which is producing keys and values as output. This communication happens through the communication attention, in which activated modules generate queries on the other modules output keys and values to read information from any other module (activated or non-activated), which helps the activated modules capture more context and other relevant information contained in all other modules.

Attention mechanisms used for selective activation of different modules and for communication between the modules are based on soft attention (Bahdanau et al., 2014; Vaswani et al., 2017; Santoro et al., 2018). Here, we generate a query Q to read from the input key K and generate an output which is a convex combination of the values V. The input attention generates queries from the hidden states of the modules on the input, and for the communication attention, the active modules generate queries on the hidden states of other modules to read relevant information and retrieve more context from them.

2.2. Meta Learning Attention

Meta-learning over a set of distributions can be interpreted as learning different types of parameters corresponding to short-term vs long-term aspects of the mechanisms underlying the generation of data. In the proposed framework, we use meta-learning to capture these structures that vary on different timescales by letting different parts of the modular network learn at different speeds. The parameters of the recurrent modules are updated more frequently to capture local variations encountered throughout the interaction, while the attention parameters which modulate connections between the modules are learnt much more slowly.

Fast and Slow Updates: The learning in the proposed framework happens in two phases: A fast learning phase updates the parameters of the recurrent modules, such that in every fast update, a subset consisting of top k most relevant modules is updated multiple times within a meta-episode to enable quick learning of the environment dynamics. Then, a second slow learning phase updates the parameters of the two attention mechanisms which lay out the connectivity structure between the modules. This helps to appropriately capture short-term (quickly changing) and long term (slowly changing) aspects of the dynamics.

3. Related Work

Meta Learning: Meta-learning (Bengio et al., 1990; Schmidhuber, 1987) gives the flexibility to adapt to new environments rapidly with a few training examples, and has demonstrated success in both supervised learning such as few shot image classification (Ravi and Larochelle, 2016) and reinforcement learning (Wang et al., 2016; Santoro et al., 2016) settings. The most relevant modular meta-learning work is that of (Alet et al., 2018), which proposes to learn modular network architecture based on MAML. The proposed work instead focuses on identifying the adaptability of each module in a given architecture.

Meta Learning to Disentangle Causal Mechanisms: Recently (Bengio et al., 2019; Ke et al., 2019) used meta learning to learn causal mechanisms or causal dependencies between a set of high level variables, inspiring the approach presented here. The 'modules' in their work are the conditional distributions for each variable in a directed graphical model and are adapted within an episode (corresponding to an intervention distribution), while the (static) connections between these modules are learnt in the outer-loop of meta-learning to form the structure of the graphical model.

4. Experiments

The experiments aim at answering the following questions: (a) Does the proposed method improve the sample efficiency, addressed positively in section 4.1. (b) Does the proposed method lead to policies that generalize better to systematic changes to the current distribution? We find positive evidence for this in section 4.2 (c) Does the proposed method lead to faster adaptation to new distributions and enable better curriculum learning to train agents in an incremental fashion that adapt faster by reusing the knowledge from their previous task? We find positive evidence in section 4.3. We also conduct ablation studies to individually disentangle the benefits of modular setup as well as meta-learning setup, summarized in section 4.4.

Environments: To answer these questions, we performed

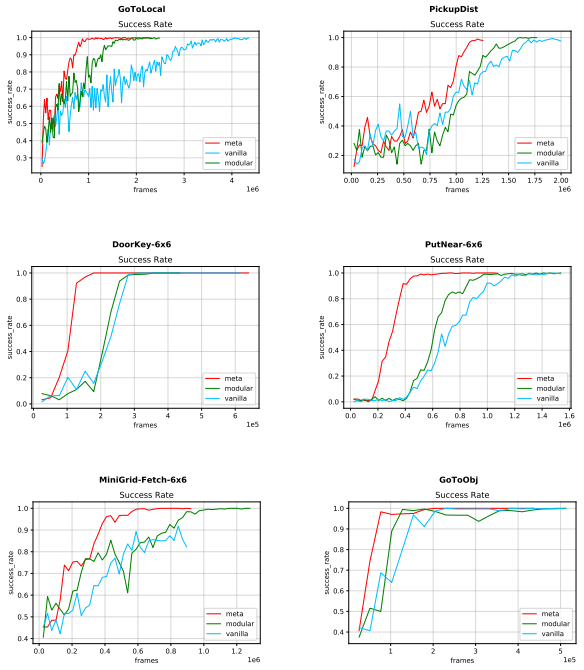

Figure 4. **Sample Efficiency on MiniGrid and BabyAI environments:** Proposed method ("meta") with modular architecture and meta-learning, outperforms the "modular" and "vanilla" setups which respectively use modular and LSTM baselines. Improvements are more profound in more difficult environments such as GoToLocal and Dynamic Obstacles.

the experiments on the MiniGrid world and BabyAI environment suite (Chevalier-Boisvert et al., 2018), which provide an agent with an egocentric, partial view of the environment. These environments are difficult for RL due to partial observability, sparse rewards, and the fact that different levels in the environment are procedurally generated.

Baselines: We compare the performance of the proposed method, referred to as "meta", with the following baselines: (a) *Vanilla LSTM* model, referred as "vanilla", (b) A modular network (i.e RIMs (Goyal et al., 2019)), called "modular", in which an ensemble of modules interact through a bottleneck of attention. In both of these baselines, all the parameters of the model are trained together and updated at the same time.

Implementation Details We used the PPO (Schulman et al., 2017) algorithm on sparse rewards such that the agent gets a positive reward only when it reaches the goal within a maximum number of n_{max} steps, and present mean-reward and average success-rate throughout our experiments.

4.1. Improved sample efficiency

One of the major issues in training reinforcement learning agents is the amount of data needed to reach human-level performance. We show how meta-learning different parameters of a modular network across different timescales helps the agent to be more sample efficient. As shown in Fig. 4, we find that the proposed method consistently improves the sample efficiency over a wide range of environments. Also the benefits of using the proposed setup becomes more evident as the environment becomes more difficultt.

4.2. Better policy generalization on Minigrid tasks

We first demonstrate that training an agent with the proposed method alone already leads to more effective policy transfer, see Fig. 4. We evaluate the capability of the proposed model to transfer knowledge from one environment (easiest) to systematically more difficult environments having some shared structure, without any further training or finetuning. If the agent has learnt the structure of the source task, it should be able to transfer knowledge to the target environments without any fine-tuning and we show positive evidence for this in Table 1 by comparing the vanilla LSTM baseline with the proposed meta attention network setup. Note that the gap in performance becomes more evident as the difficulty of the environment increases.

Table 1. **Zero shot Policy Transfer:** The model is trained on the easiest environment, and transferred in a zero-shot manner to more difficult and larger environments, winning over the LSTM baseline. $R(.)$ and $S(.)$ represent the Mean Reward and Success Rates respectively.

Target Environment	R(Meta Attention)	R(LSTM)	S(Meta Attention)	S(LSTM)
Easy	0.821	0.634	0.876	0.689
Medium	0.715	0.362	0.764	0.394
Difficult	0.424	0.086	0.452	0.104

4.3. Efficient pre-training and knowledge transfer for curriculum learning

Here, we demonstrate that training an agent with the proposed method alone already leads to a more effective policy that improves the sample efficiency for the downstream task (such as in Curriculum learning). To evaluate how well the proposed method transfers knowledge from the previous tasks, we use models pretrained on easier environments to train on more difficult ones which have some shared structure with the source environment. In Fig. 5(a), we find that the proposed model adapts better, as compared to a LSTM setup, thus showing that the proposed model is able to use the past experience more efficiently by not having to update all the parameters and relearn everything for the new task.

4.4. Ablation analysis: benefits of meta-learning setup

To understand what is benefiting the proposed method, we performed an ablation in which we trained an LSTM baseline in a similar meta-learning fashion, referred as meta-LSTM. For this, the parameters of the LSTM and the agent's policy are learnt in the inner loop, and the parameters of

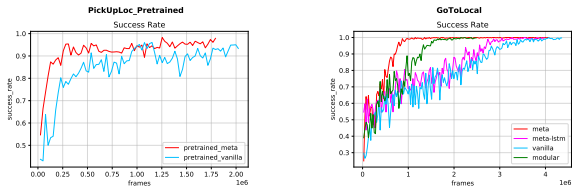

Figure 5. **Knowledge for Curriculum Learning and Ablation study experiments**: (a) The proposed model is trained on GoToLocal source environment, and then fine-tuned on a more difficult environment (PickUpLoc) that shares some structure and competencies with the source environment, and outperforms the LSTM baseline pretrained in similar way. (b) Importance of both meta learning and modularization in the proposed method is evident as an LSTM baseline trained in a similar meta-learning fashion performs better than the vanilla version. The proposed method still outperforms all other setups

value function are meta-learnt in the outer loop. We show that separating the learning of the parameters of the policy and value function into two timescales using this meta-learning setup improves over the LSTM baseline, as shown in Fig. 5(b), highlighting the importance of both modular architecture and meta-learning in the proposed setup.

5. Conclusion

This paper investigates using a meta-learning approach on modular architectures to capture short-term vs long-term aspects of the underlying mechanisms in the data generation process, by considering parameters of attention mechanism as meta-parameters, and parameters of the recurrent modules as parameters. The experimental results on grounded language learning tasks in the RL setting strongly indicate that the combination of meta-learning of the attention parameters and dynamically connected modular architectures with sparse communication leads in many ways to superior results in terms of improved sample efficiency (faster convergence, higher mean return and success rates), and an improved transfer across tasks in a curriculum, both as zero-shot transfer and with adaptation. Ablation studies further confirm that using a meta-learning approach to update different parameters of the network over different timescales leads to improvements in sample efficiency as compared to training all the parameters at once. We also show that using only a modular architecture, or only meta learning on a standard monolithic architecture do not perform as well as the proposed method. Overall, the results point towards a novel way to perform meta-learning and attention-based modularization for better sample efficiency, out-of-distribution generalization and transfer learning in RL.

References

Ferran Alet, Tomás Lozano-Pérez, and Leslie P Kaelbling. Modular meta-learning. *arXiv preprint arXiv:1806.10166*, 2018.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Citeseer, 1990.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv:1901.10912*, 2019.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: First steps towards grounded language learning with a human in the loop. *arXiv preprint arXiv:1810.08272*, 2018.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

Adam Santoro, Ryan Faulkner, David Raposo, Jack W. Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy P. Lillicrap. Relational recurrent neural networks. *CoRR*, abs/1806.01822, 2018. URL http://arxiv.org/abs/1806.01822.

Jurgen Schmidhuber. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, 1:2, 1987.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.