Knowledge-Based Systems xxx (xxxx) xxx

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Prior-knowledge and attention based meta-learning for few-shot learning

Yunxiao Qin [a,*], Weiguo Zhang [a], Chenxu Zhao [b], Zezheng Wang [c], Xiangyu Zhu [d], Jingping Shi [a], Guojun Qi [e], Zhen Lei [d,f]

[a] Northwestern Polytechnical University, Xian, 710129, China
[b] MiningLamp Technology, Beijing, 100094, China
[c] Beijing Kwai Technology, Beijing, 102600, China
[d] National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Science, Beijing 100000, China
[e] Huawei Cloud, Seattle 90876, USA
[f] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

*Article history:*
Received 27 January 2020
Received in revised form 9 October 2020
Accepted 12 November 2020
Available online xxxx

*Keywords:*
Meta-learning
Few-shot learning
Prior-knowledge
Representation
Attention mechanism

ABSTRACT

Recently, meta-learning has been shown to be a promising way to solve few-shot learning. In this paper, inspired by the human cognition process, which utilizes both prior-knowledge and visual attention when learning new knowledge, we present a novel paradigm of meta-learning approach that capitalizes on three developments to introduce attention mechanism and prior-knowledge to meta-learning. In our approach, prior-knowledge is responsible for helping the meta-learner express the input data in a high-level representation space, and the attention mechanism enables the meta-learner to focus on key data features in the representation space. Compared with the existing meta-learning approaches that pay little attention to prior-knowledge and visual attention, our approach alleviates the meta-learner's few-shot cognition burden. Furthermore, we discover a Task-Over-Fitting (TOF) problem,[1] which indicates that the meta-learner has poor generalization across different $K$-shot learning tasks. To model the TOF problem, we propose a novel Cross-Entropy across Tasks (CET) metric.[2] Extensive experiments demonstrate that our techniques improve the meta-learner to state-of-the-art performance on several few-shot learning benchmarks while also substantially alleviating the TOF problem.



## 1. Introduction

The development of deep learning has led to remarkable advancements in many computer vision tasks [1–3]. Deep learning-based approaches usually require thousands or even millions of labeled samples to obtain satisfactory performance. However, collecting and annotating such enormous data is notoriously expensive. Therefore, few-shot learning [4–6], which requires deep networks to learn from a few data, becomes a hotspot problem in Computer Vision.

Learning from limited data is challenging for deep learning based image classification. In comparison, we human beings can rapidly learn new categories from very few examples. Recently, meta-learning [7–18] has shown promising performance to improve few-shot learning for Computer Vision. However, the existing meta-learning methods commonly ignore prior-knowledge [19–23] and attention mechanism [24,25] which have both been demonstrated to be important for human cognitive and learning processes. Fig. 1 illustrates a few-shot classification problem to provide a clearer understanding of the role of prior-knowledge and attention mechanism in human few-shot learning. In Fig. 1, we unconsciously leverage our learned knowledge about the world to understand and express these images into high-level compact representations, such as plant, animal, tree, and table *etc.* However, according to the four training images, we discover that only the features of the tree and table are useful for us to recognize these two image classes. Finally, we quickly adjust ourselves to pay attention to the critical features and make a precise decision based on the focused features.

* Corresponding author.
*E-mail addresses:* qyxqyx@mail.nwpu.edu.cn (Y. Qin), zhangwg@nwpu.edu.cn (W. Zhang), zhaochenxu@mininglamp.com (C. Zhao), wangzezheng@kuaishou.com (Z. Wang), xiangyu.zhu@nlpr.ia.ac.cn (X. Zhu), shijingping@nwpu.edu.cn (J. Shi), guojunq@gmail.com (G. Qi), zlei@nlpr.ia.ac.cn (Z. Lei).

[1] When tested on $J$-shot classification tasks, the meta-learner trained on $K$-shot tasks does not perform as well as the one trained on $J$-shot tasks, where $K$ and $J$ are different unsigned integers denoting different numbers of shots for the meta-learner.

[2] A metric for quantizing the extent to which a meta-learning method suffers from the TOF problem.

https://doi.org/10.1016/j.knosys.2020.106609
0950-7051/

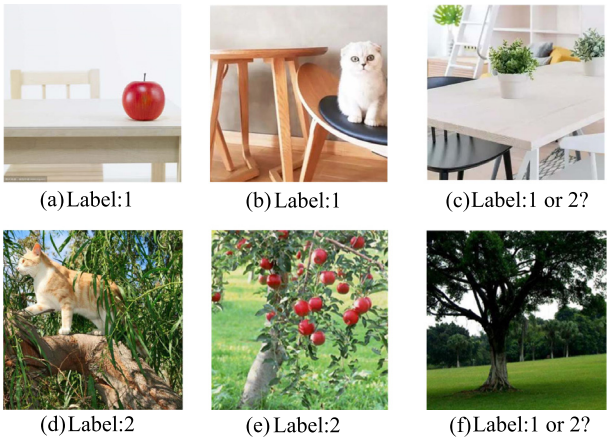


**Fig. 1.** An example of few-shot classification task. The six images belong to two classes; the four labeled images are training data, and the two unlabeled images are testing data. When predicting the two test images, we utilize our prior-knowledge about the world to understand all components in these images and use visual attention to pay attention to the key components—table and tree. Finally, we predict that image (c) belongs to class 1 which contains table, while image (f) is associated with class 2 which contains tree.

Evidently, we can summarize two main modules in human few-shot learning: **a stable Representation module that utilizes prior-knowledge to express the image into compact feature representations; and a smart attention-based logical decision module that adapts accurately and performs recognition based on the feature representations**. Whereas, the existing meta-learning approaches commonly train meta-learners to learn adaptive networks directly based on the original input data with neither attention mechanism nor prior-knowledge.

In this paper, inspired by human cognition, we present a novel paradigm of meta-learning approach with three developments to introduce attention mechanism and prior-knowledge to meta-learning in a step-by-step fashion. Here, we briefly introduce the proposed methods. **(1)** The first method is called **A**ttention-based **M**eta-**L**earning (AML). It leverages attention mechanism to enable the meta-learner to pay more attention to critical features. **(2)** To enable the meta-learner enjoying not only attention but also prior-knowledge, we present another method **R**epresentation and **A**ttention based **M**eta-**L**earning (RAML). Its network contains a Representation module and an attention-based prediction (ABP) module. The Representation module is similar to the Representation module of human vision. It learns the prior-knowledge in a supervised fashion and is responsible for understanding and extracting stable compact feature representations from the input image. The ABP module plays the same role as the smart attention-based logic decision module of human vision. It enables the meta-learner to precisely adjust first its attention to the most discriminative feature representations of input images and second its predictions. **(3)** In the third method, to take full advantage of endless unlabeled data, the Representation module learns the prior-knowledge through an unsupervised learning process [26–32]. We call this method **U**nsupervised **R**epresentation and **A**ttention based **M**eta-**L**earning (URAML). With URAML, our experiments show that the growth in the number of unlabeled data and the development of unsupervised learning both improve the performance of URAML apparently.

In addition, we show the existence of a Task-Over-Fitting (TOF) problem for the existing meta-learning methods, and present a Cross-Entropy across Tasks (CET) metric to evaluate how much a meta-learning method is troubled by the TOF problem. An example of the TOF problem is that the meta-learner trained on 5-way 1-shot tasks is not as capable as the one trained on 5-way 5-shot tasks when both of them are tested on 5-way 5-shot tasks, and vice versa. However, in practical applications, it is uncertain how much data and how many shots are available to the meta-learner to learn. Therefore, we argue that the trained meta-learner should generalize well to different $K$-shot tasks. The possible underlying reason for the TOF problem is that the existing meta-learning methods ignore prior-knowledge, which results in the feature extractor of their meta-learners overfit to the training $K$-shot tasks. Moreover, ignoring attention mechanism makes the existing meta-learners be vulnerable to interference from features irrelevant to the presented tasks. Our experiment validates that by incorporating prior-knowledge and attention mechanism, our methods suffer less from the TOF problem than do the existing meta-learning methods.

We summarize the main contributions of our work as follows.

- We argue that attention mechanism and prior-knowledge are both crucial for meta-learners to reduce their cognition burden in few-shot learning. To validate our viewpoint, we develop a novel paradigm with three methods AML, RAML, and URAML to leverage attention mechanism and prior-knowledge in meta-learning.
- We discover the TOF problem for meta-learning and design a novel metric Cross-Entropy across Tasks (CET) to measure the extent to which meta-learning approaches suffer from the TOF problem.
- Through extensive experiments, we show that the proposed methods achieve state-of-the-art performance on several few-shot learning benchmarks. Meanwhile, compared with the existing meta-learning methods, they are also less sensitive to the TOF problem, particularly RAML and URAML.

## 2. Related work

### 2.1. Meta-learning for few-shot learning

Few-shot learning tasks are also called as $N$-way $K$-shot learning tasks. Each $N$-way $K$-shot task contains a support set and a query set [4,10]. The support and query set contain $K$ and $L$ examples for each of the $N$ classes, respectively. The existing meta-learning approaches [10–18,33–38] usually solve the few-shot learning by training a meta-learner on the $N$-way $K$-shot learning tasks in the following way. Firstly, the meta-learner is required to inner-update itself on the support set. Secondly, after inner-updating, the meta-learner is evaluated on the query set. Finally, by minimizing the loss on the query set, the meta-learner learns a base learner that has easy-fine-tune weights [10,13,37,39] or a skillful weight updater [12,18] or both [11] or the ability to memorize the support set [14]. The methods which train the meta-learner learning an easy-fine-tune base learner are also called as weight initialization based methods, because the meta-learner learns generalized initial weight for few-shot learning tasks. Recently, MAML, a classical weight initialization based method, is popular and lots of MAML based methods have been proposed. For example, LLAML [34] uses a local Laplace approximation to model the task parameters, and MTL [40] trains a meta-transfer to adapt a pre-trained deep network to few-shot learning tasks. Besides, MetaGAN [17] shows that by coupling MAML with adversarial training, the meta-learner is trained to learn better decision boundaries between different classes in few-shot learning. To reduce the computation and memory cost of MAML, iMAML [33] leverages implicit differentiation to remove the need of differentiation through the inner-update path.

Though the existing meta-learning methods perform promising, they seldom consider prior-knowledge and attention mechanism in meta-learning. In this paper, we improve meta-learning for few-shot learning by introducing prior-knowledge and attention mechanism to meta-learning.

### 2.2. Attention mechanism

Recent years, attention mechanism [41–44] has been widely used in computer vision systems, machine translation, and natural language processing systems. Several manners of the attention mechanism have been proposed, such as soft attention [41,42], hard attention [43], and self attention [44]. Soft attention trains a weight mask for the hidden features and calculates the attentive features by multiplying the weight mask with the hidden features. The features which multiples with larger weight will be focused on by the deep model. SENet [42] takes advantage of soft attention mechanism to win the champion on the image classification task of ILSVRC-2017 [45]. Hard attention [43] can be seen as a module that decides an image block region that is visible to the network, and the other regions are invisible to the network. Self-attention [44] improves the performance of the machine translation system by training a network to find the inner dependency of the input and that of the output. In this paper, we use soft attention mechanism as the meta-learner's attention mechanism.

### 2.3. Unsupervised representation learning

Supervised learning is data-hungry at training deep networks, which costs a great deal of manual annotation effort. Considering this disadvantage of supervised learning, several unsupervised learning approaches [26,27,29–32,46] have been proposed. The traditional unsupervised learning way is training a neural network to reconstruct the input through an Encoder–Decoder architecture, such as Auto-Encoder [26] and Variational Auto-Encoder (VAE) [27]. Colorization [29] uses *Lab* images to train an encoder–decoder network to predict the unseen *ab* channels based only on the input *L* channel. Split-Brain [31] extend Colorization by training two separated encoder–decoder networks to independently predict (1) the *ab* channels based on the *L* channel and (2) predict the *L* channel based on the *ab* channels. Except for the above methods, image rotation based methods [46,47] present another direction for unsupervised learning. For instance, RotNet [46] trains a network to predict the rotation of the randomly rotated input image. Recently, contrastive learning based methods [32,48] develop quickly and show their promising unsupervised learning performance. In this work, Split-Brain is default utilized for URAML learning the prior-knowledge about image classification. Besides, in our experiment, we show that URAML is compatible with not only Split-Brain but also the other unsupervised learning methods, such as RotNet [46] and MoCo [32].

## 3. Methodology

### 3.1. Problem of learning from few data

Learning from limited data is extremely difficult for deep learning models. One reason is that the original input data are commonly represented in a large dimensional space, typically with tens or hundreds of thousands of dimensions. For example, for the image classification task, each original image is commonly stored in a large dimensional space (the dimension of a $224 \times 224$ RGB image is 150,528). In this large dimensional space, it is difficult for a few samples of one category to accurately reflect the entire characters of this category.

Humans are able to learn new categories efficiently because they utilize prior-knowledge and attention mechanism in cognition [19,20,22,23,49–53]. Prior-knowledge allows humans to express perceptual images as high-level representations or descriptions, while attention mechanism enables humans to focus on the critical components of the representations. In this way, humans reduce the dimension of images while maintaining the discriminative image components. This process alleviates the human cognition load and facilitates humans to learn new categories efficiently.

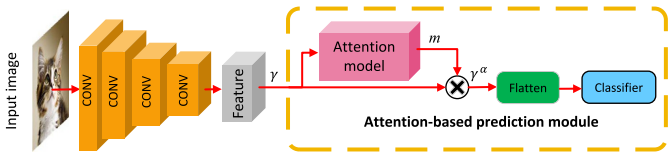


**Fig. 2.** Network structure of the proposed AML.

The existing meta-learning methods [10,11,13,16,33,34] have greatly improved deep learning in learning from few data. However, they focus mainly on training the meta-learner to quickly adapt its network to fit few-shot learning tasks directly based on the few original high-dimensional input data. Prior-knowledge and attention mechanism were almost ignored in these methods, leading to unsatisfactory performances. Besides, as introduced before, we propose that ignoring prior-knowledge and attention mechanism is also the possible reason why the existing meta-learning approaches are vulnerable to suffer from the TOF problem.

In this paper, inspired by human cognition and for addressing the problem the existing meta-learning approaches expose, we propose three methods in a step-by-step manner: Attention based Meta-Learning (AML), Representation and Attention based Meta-Learning (RAML), Unsupervised Representation and Attention based Meta-Learning (URAML).

### 3.2. AML

AML equips the meta-learner with the ability to control its attention. We first introduce the network structure and then detail the training of AML.

#### 3.2.1. Network of AML

Fig. 2 shows the network architecture of AML. The network consists of a feature extractor and an attention-based prediction (ABP) module. The feature extractor is a CNN $\mathcal{F}$ composed of four cascaded convolutional layers. The ABP module contains a convolution-based attention model $\mathcal{A}$ and a fully-connect layer-based classifier $\mathcal{C}$. Eq. (1) shows the inference of the network. $\theta_f$, $\theta_a$, and $\theta_c$ are the weights of $\mathcal{F}$, $\mathcal{A}$, and $\mathcal{C}$, respectively. $\mathcal{F}$ extracts features $\gamma_i$ from the input image $x_i$. Then, $\mathcal{A}$ calculates the soft attention mask $m_i$ for the features $\gamma_i$. By channel-wise multiplication $\odot$ between $\gamma_i$ and $m_i$, the attentive features $\gamma_i^\alpha$ are obtained. Finally, the classifier $\mathcal{C}$ outputs the prediction $\hat{y}_i$ for the image $x_i$. We simplify and integrate the overall inference in Eq. (1) as $\hat{y}_i = \mathbb{F}(x_i; \theta_f, \theta_a, \theta_c)$.

$$\begin{cases} \gamma_i &= \mathcal{F}(x_i;\ \theta_f) \\ m_i &= \mathcal{A}(\gamma_i;\ \theta_a) \\ \gamma_i^\alpha &= \gamma_i \odot m_i \\ \hat{y}_i &= \mathcal{C}(\gamma_i^\alpha;\ \theta_c) \end{cases} \tag{1}$$

In this paper, we use soft attention mechanism to build the attention model. Although soft attention does not function exactly the same as the attention mechanism in human vision, it plays a similar role—it helps the meta-learner control its attention to key features.

Fig. 3 shows the attention model structure. The input feature $\gamma$ is first global-average-pooled to obtain the feature $\gamma'$. Then, a convolution layer and a sigmoid layer are used to predict the

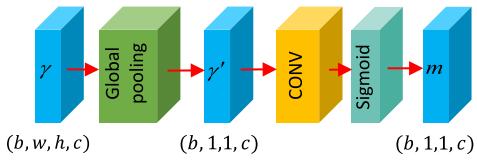


**Fig. 3.** The inner structure of the attention model. The shape of feature map $\gamma$ is (*b*,*w*,*h*,*c*) (at the left of the figure), where *b, w, h, c* are the batch size, width, height and number of feature map channels of $\gamma$. The shapes of both $\gamma'$ and $m$ are (*b*,1,1,*c*)..

attention mask $m$ from the feature $\gamma'$. We formulate the attention model in Eq. (2).

$$\begin{cases} \gamma' = \mathcal{P}_a(\gamma), \\ m = \sigma(\mathcal{F}_a(\gamma';\ \theta_a)) \end{cases} \tag{2}$$

$\mathcal{P}_a$ is the global-average pooling operation, $\sigma$ is the sigmoid activation, and $\mathcal{F}_a$ is the convolution layer in the attention model.

### 3.2.2. Meta-training of AML

Given a few-shot classification task $\tau$, AML meta-trains the meta-learner to solve the task $\tau$ through the following two steps.

**First**, AML requires the meta-learner to inner-update itself on the support set of $\tau$, which can be formulated as Eqs. (3) and (4).

$$\begin{cases} \hat{y}_i = \mathbb{F}(x_i; \theta_f, \theta_a, \theta_c), \\ \mathcal{L}_i(\theta_f, \theta_a, \theta_c) = l(\hat{y}_i, y_i), \\ \mathfrak{L}_s(\theta_f, \theta_a, \theta_c) = \frac{1}{N_s}\sum_{i=1}^{N_s}\mathcal{L}_i(\theta_f, \theta_a, \theta_c) \end{cases} \tag{3}$$

$$(\theta_f', \theta_a', \theta_c') = (\theta_f, \theta_a, \theta_c) - \alpha \circ \nabla_{(\theta_f, \theta_a, \theta_c)}\mathfrak{L}_s(\theta_f, \theta_a, \theta_c) \tag{4}$$

In Eq. (3), $x_i$ is the $i$th image of the support set. $l$ is the cross-entropy loss function and $\mathcal{L}_i$ is the meta-learner's loss on the image $x_i$. $\mathfrak{L}_s$ is the meta-learner's loss on the total support set and $N_s$ is the number of images in the support set. In Eq. (4), inspired by Meta-SGD [11], we set $\alpha$ to a trainable vector which adjusts the inner-update direction and step size. $\alpha$ can also be detailed as $\alpha = [\alpha_f, \alpha_a, \alpha_c]$. $\alpha_f$, $\alpha_a$, and $\alpha_c$ have the same shape as the weights $\theta_f$, $\theta_a$, and $\theta_c$, respectively. Therefore, Eq. (4) can be split into three equations, *i.e.* $\theta_f' = \theta_f - \alpha_f \circ \nabla_{\theta_f}\mathfrak{L}_s(\theta_f, \theta_a, \theta_c)$ and *etc.*. For simplicity, we merge these three equations into one equation as Eq. (4) shows. $\circ$ is the element-wise multiplication. With Eqs. (3) and (4), the meta-learner inner-updates its weights $\theta_f, \theta_a, \theta_c$ to $\theta_f', \theta_a', \theta_c'$.

**Second**, because the inner-updated weight $\theta_f', \theta_a'$, and $\theta_c'$ depend not only on the initial values of $\theta_f, \theta_a$, and $\theta_c$, but also on $\alpha$, all $\theta_f, \theta_a, \theta_c$, and $\alpha$ can be meta-optimized. We formulate the meta-optimization as Eqs. (5) and (6).

$$\begin{cases} \hat{y}_i = \mathbb{F}(x_i; \theta_f', \theta_a', \theta_c'), \\ \mathcal{L}_i(\theta_f', \theta_a', \theta_c') = l(\hat{y}_i, y_i), \\ \mathfrak{L}_q(\theta_f', \theta_a', \theta_c') = \frac{1}{N_q}\sum_{i=1}^{N_q}\mathcal{L}_i(\theta_f', \theta_a', \theta_c') \end{cases} \tag{5}$$

$$(\theta_f, \theta_a, \theta_c, \alpha) = (\theta_f, \theta_a, \theta_c, \alpha) - \beta \cdot \nabla_{(\theta_f, \theta_a, \theta_c, \alpha)}\mathfrak{L}_q(\theta_f', \theta_a', \theta_c') \tag{6}$$

In Eq. (5), $x_i$ is the $i$th image of the query set, and $N_q$ denotes the number of images in the query set. $\mathfrak{L}_q$ is the inner-updated meta-learner's loss on the query set. Note that $\nabla_{(\theta_f, \theta_a, \theta_c, \alpha)}\mathfrak{L}_q(\theta_f', \theta_a', \theta_c')$ computes the gradient of $\mathfrak{L}_q$ towards $(\theta_f, \theta_a, \theta_c, \alpha)$ but not $(\theta_f', \theta_a', \theta_c')$. By optimizing $\mathfrak{L}_q$, the meta-learner is forced to learn not only the suitable initial weights $\theta_f, \theta_a, \theta_c$ but also $\alpha$ for task $\tau$. With the learned initial weights and $\alpha$, the meta-learner can inner-update itself precisely on the support set and then perform well on the query set.

In AML, the meta-learner is trained on multiple few-shot learning tasks using these two steps. This process causes the meta-learner to learn generalizable initial weights not only for the feature extractor $\mathcal{F}$ and the classifier $\mathcal{C}$ but also for the attention model $\mathcal{A}$. In contrast, the existing initialization-based meta-learning methods train the meta-learner to learn initial weights for only the feature extractor and the classifier. Therefore, compared with the existing meta-learners, AML simplifies the few-shot learning problem and improves the performance since it learns how to quickly focus its attention on the features that are crucial to solving few-shot learning. In our experiments, we validate the effectiveness of this attention mechanism.

## 3.3. RAML

RAML assembles the meta-learner with not only the attention mechanism but also the ability to capitalize on the learned past knowledge.

### 3.3.1. Structure of RAML

Fig. 4 shows the meta-learner's network structure. It consists of a Representation module and an ABP module. The Representation module differs from the feature extractor in AML; it is responsible for the meta-learner leveraging prior-knowledge to understand the input image. Whereas, the feature extractor in AML is meta-trained to be responsible for quickly adjusting itself to solve few-shot learning tasks. In our work, the Representation module is a ResNet-50 network. Similar to the ABP module in AML, the ABP module in RAML also contains an attention model and a classifier. It is responsible for quickly adjusting the meta-learner's attention and prediction based on the compact high-level features extracted by the Representation module. Note that the Auxiliary module shown in Fig. 4 does not belong to the meta-learner and it is used only to assist the meta-learner learning prior-knowledge.

### 3.3.2. Training of RAML

The training process of RAML can be separated into two stages: prior-knowledge learning and meta-training stages.

**At the prior-knowledge learning stage**, with the assistance of the Auxiliary module, the Representation module is trained to learn prior-knowledge about image classification in a supervised manner. The training process can be formulated as

$$\begin{cases} \gamma_i = \mathcal{F}_r(x_i;\ \theta_r) \\ \hat{y}_i = \mathcal{C}_{au}(\gamma_i;\ \theta_{au}) \\ \theta_r^*, \theta_{au}^* = \underset{\theta_r, \theta_{au}}{argmin}\frac{1}{n}\sum_{i=1}^{n} l(\hat{y}_i, y_i) \end{cases} \tag{7}$$

$\mathcal{F}_r$ and $\mathcal{C}_{au}$ denote the Representation and Auxiliary modules, respectively, and $\theta_r$ and $\theta_{au}$ are their respective weights. $x_i$ is an input image used for the representation model learning prior-knowledge, and $n$ is the number of images. $\theta_r^*$ and $\theta_{au}^*$ are the learned values of $\theta_r$ and $\theta_{au}$.

**At the meta-training stage**, we train the meta-learner on amount of few-shot learning tasks. For the meta-learner can use the learned knowledge well to stably express the input image in high-level representation space, we freeze the Representation module and meta-train only the ABP module. Similar to AML, in RAML, we simplify the meta-learner's prediction as $\hat{y}_i = \mathbb{F}(x_i; \theta_r^*, \theta_a, \theta_c)$, where $\theta_r^*$ is the Representation module's learned weight at the prior-knowledge stage. $\theta_a$ and $\theta_c$ respectively denote the weights of the attention model and classifier in the ABP module.

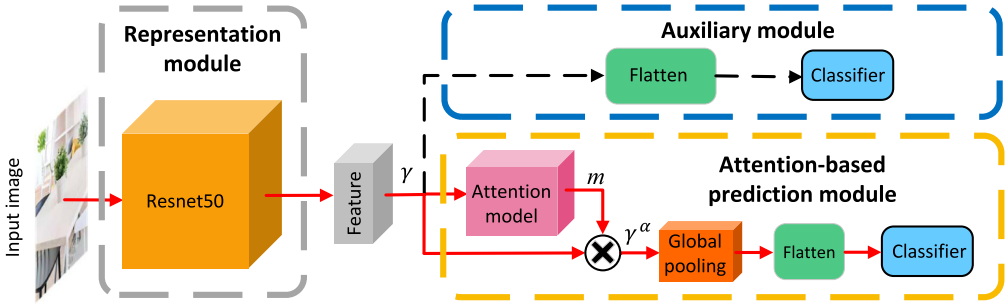


**Fig. 4.** The network structure of the proposed RAML. The meta-learner is composed of a Representation module and an ABP module. The Auxiliary module is used to assist the meta-learner to learn prior-knowledge.

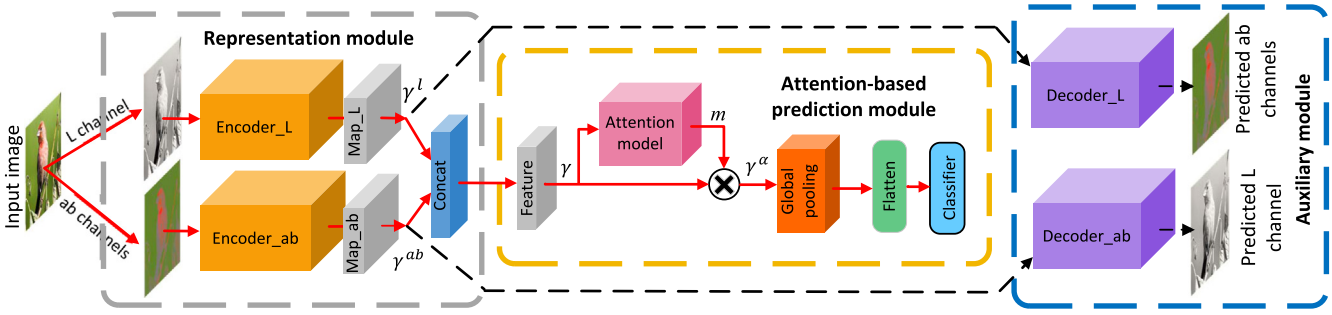


**Fig. 5.** The network structure of URAML. The meta-learner is composed of a Representation module and an ABP module. The Auxiliary module is used to assist the meta-learner to learn prior-knowledge. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Given a few-shot learning task, we formulate the inner-update of the meta-learner on the corresponding support set as Eqs. (8) and (9). Unlike the inner-update of AML which updates all network weights, RAML only inner-updates the weights $\theta_a$ and $\theta_c$ of the ABP module. The Representation module weight $\theta_r^*$ is frozen to avoid forgetting the learned prior-knowledge.

$$\begin{cases} \hat{y}_i = \mathbb{F}(x_i; \theta_r^*, \theta_a, \theta_c), \\ \mathcal{L}_i(\theta_r^*, \theta_a, \theta_c) = l(\hat{y}_i, y_i), \\ \mathfrak{L}_s(\theta_r^*, \theta_a, \theta_c) = \frac{1}{N_s}\sum_{i=1}^{N_s}\mathcal{L}_i(\theta_r^*, \theta_a, \theta_c) \end{cases} \tag{8}$$

$$(\theta_a', \theta_c') = (\theta_a, \theta_c) - \alpha \circ \nabla_{(\theta_a, \theta_c)}\mathfrak{L}_s(\theta_r^*, \theta_a, \theta_c) \tag{9}$$

After inner-updating, RAML meta-trains the meta-learner on the corresponding query set, which can be formulated as Eqs. (10) and (11).

$$\begin{cases} \hat{y}_i = \mathbb{F}(x_i; \theta_r^*, \theta_a', \theta_c'), \\ \mathcal{L}_i(\theta_r^*, \theta_a', \theta_c') = l(\hat{y}_i, y_i), \\ \mathfrak{L}_q(\theta_r^*, \theta_a', \theta_c') = \frac{1}{N_q}\sum_{i=1}^{N_q}\mathcal{L}_i(\theta_r^*, \theta_a', \theta_c') \end{cases} \tag{10}$$

$$(\theta_a, \theta_c, \alpha) = (\theta_a, \theta_c, \alpha) - \beta \cdot \nabla_{(\theta_a, \theta_c, \alpha)}\mathfrak{L}_q(\theta_r^*, \theta_a', \theta_c') \tag{11}$$

The main characteristic of RAML is that the Representation module and the ABP module are trained separately. The Representation module is trained in a supervised manner to learn the prior-knowledge about image classification, while the ABP module is meta-trained to learn how to adjust itself quickly to solve few-shot learning tasks in the representation space provided by the Representation module. Compared with AML, which meta-trains the meta-learner not only adjusting the feature extractor but also the ABP module, RAML simplifies the few-shot learning problem because the meta-learner needs to adjust only its ABP module when solving few-shot learning tasks. This may explain why RAML outperforms AML in our experiment.

### 3.4. URAML

Prior-knowledge can be learned not only from labeled data but also from large-scale unlabeled data. Thus, we develop the method URAML which learns the prior-knowledge in an unsupervised learning manner. Fig. 5 shows its network structure. Similar to RAML, the meta-learner is composed of a Representation module and an ABP module. The Auxiliary module in Fig. 5 does not belong to the meta-learner. The training process of URAML can also be separated into two stages: prior-knowledge learning and meta-training stages.

**At the prior-knowledge learning stage**, the Representation module learns the knowledge with an unsupervised learning method Split-Brain [31]. Split-Brain simultaneously trains two encoder–decoders on *Lab* images. In *Lab* color system, the *L* channel determines the image brightness, and the *ab* channels determine the image color. One encoder–decoder in Split-Brain is trained to predict the unseen *ab* channels of the input *Lab* image, given only the *L* channel. Another is trained to predict the unseen *L* channel, given the *ab* channels. As Fig. 5 shows, the Representation module consists of two ResNet-50 based encoders and the Auxiliary module consists of two deconvolution [54] based decoders. We formulate the prior-knowledge learning process as Eqs. (12) and (13).

$$\begin{cases} \gamma_i^l = \mathcal{F}_l(x_i^l;\ \theta_l) \\ \hat{x}_i^{ab} = \mathcal{D}_l(\gamma_i^l;\ \omega_l) \\ L_l(\theta_l, \omega_l) = \frac{1}{n}\sum_{i=1}^{n} l_2(x_i^{ab},\ \hat{x}_i^{ab}) \\ \theta_l^*, \omega_l^* = \underset{\theta_l, \omega_l}{argmin} L_l(\theta_l, \omega_l) \end{cases} \tag{12}$$

In Eq. (12), $x_i^l$ and $x_i^{ab}$ are the *L* and *ab* channels of the input *Lab* image $x_i$, respectively. $\mathcal{F}_l$ and $\mathcal{D}_l$ are the encoder and decoder, respectively, and $\theta_l$ and $\omega_l$ are their corresponding weights. The encoder extracts the features $\gamma_i^l$ from $x_i^l$ and the decoder predicts

$\hat{x}_i^{ab}$ based on the features $\gamma_i^l$. $\theta_l^*$ and $\omega_l^*$ are the optimized values of $\theta_l$ and $\omega_l$. $L_l$ is the loss of $\mathcal{F}_l$ and $\mathcal{D}_l$, and $l_2$ is the mean squared error ($MSE$) loss function. $n$ is the number of $Lab$ images used to train $\mathcal{F}_l$ and $\mathcal{D}_l$. Similarly, we formulate another encoder–decoder pair which predicts $L$ channel from $ab$ channels as Eq. (13).

$$\begin{cases} \gamma_i^{ab} = \mathcal{F}_{ab}(x_i^{ab};\ \theta_{ab}) \\ \hat{x}_i^l = \mathcal{D}_i^{ab}(\gamma_i^{ab};\ \omega_{ab}) \\ L_{ab}(\theta_{ab}, \omega_{ab}) = \frac{1}{n}\sum_{i=1}^{n} l_2(x_i^l,\ \hat{x}_i^l) \\ \theta_{ab}^*, \omega_{ab}^* = \underset{\theta_{ab},\omega_{ab}}{argmin} L_{ab}(\theta_{ab}, \omega_{ab}) \end{cases} \tag{13}$$

After unsupervised learning, the representations $\gamma_i$ of an $Lab$ image $x_i$ can be obtained with the formulation $\gamma_i = Con(\gamma_i^l, \gamma_i^{ab})$, where $Con()$ is the concatenate operation.

**At the meta-training stage**, the ABP module is trained in the same way as RAML. Note that, the learned weight of the Representation module in URAML is denoted as $\theta_r^* = [\theta_l^*, \theta_{ab}^*]$.

Here, we summarize our three methods briefly. Inspired by human cognition, which makes full use of attention mechanism and prior-knowledge to efficiently learn new knowledge, we design a novel paradigm by capitalizing on three methods to introduce attention mechanism and prior-knowledge to meta-learning in a step-by-step manner. First, the method AML is designed to leverage attention mechanism in meta-learning. Second, the method RAML is designed to use both the attention mechanism and prior-knowledge in meta-learning. Unlike RAML, URAML learns prior-knowledge through unsupervised learning, which confers the advantage that the meta-learner performance can be boosted with (1) the growth of the number of unlabeled images used at the prior-knowledge learning stage and (2) the progress of the unsupervised learning algorithm.

## 4. Experiments

### 4.1. Experimental dataset

We use several datasets in all our experiments: MiniImagenet [12], Omniglot [55], MiniImagenet-900, Places2 [56], COCO2017 [57], and OpenImages-300. Note that, we resize all the images in Omniglot into 28 × 28 resolution, and all the other images into 84 × 84.

#### 4.1.1. MiniImagenet

MiniImagenet [12] is popularly used for evaluating few-shot learning and meta-learning. It contains 100 image classes, including 64 training classes, 16 validation classes, and 20 testing classes. Each image class with 600 images are sampled from the ImageNet dataset [58].

#### 4.1.2. Omniglot

Omniglot [55] is another widely used dataset for few-shot learning. It contains 50 different alphabets and 1623 characters from these alphabets, and each character has 20 images that hand-drawn by 20 different people.

#### 4.1.3. MiniImagenet-900

MiniImagenet-900 dataset is designed for the Representation modules in RAML and URAML learning prior-knowledge. It contains 900 image classes and each image class with 1300 images are collected from the original ImageNet dataset. It is worth noting that there is no image class in MiniImageNet-900 coincides with the classes in the MiniImagenet dataset.

**Table 1**
Few-shot learning performance on MiniImagenet. The methods shown in the upper part of the table use shallow networks to extract image features while those shown in the lower part use deep networks. All proposed methods are presented in bold font.

| Method | Venue | 5-way accuracy | |
|---|---|---|---|
| | | 1-shot | 5-shot |
| MAML [10] | ICML-17 | 48.70 ± 1.84% | 63.11 ± 0.92% |
| Prototypical nets [5] | NIPS-17 | 49.42 ± 0.78% | 68.20 ± 0.66% |
| Meta-SGD [11] | / | 50.47 ± 1.87% | 64.03 ± 0.94% |
| LLAMA [34] | ICLR-18 | 49.40 ± 1.83% | / |
| Relation Net [60] | CVPR-18 | 51.38 ± 0.82% | 67.07 ± 0.69% |
| GNN [61] | ICLR-18 | 50.33 ± 0.36% | 66.41 ± 0.63% |
| Spot-Learn [62] | CVPR-19 | 51.03 ± 0.78% | 67.96 ± 0.71% |
| iMAML HF [33] | NIPS-19 | 49.30 ± 1.88% | / |
| Meta-MinibatchProx [63] | NIPS-19 | 50.77 ± 0.90% | 67.43 ± 0.89 |
| NIL [39] | ICLR-20 | 48.00 ± 0.70% | 62.20 ± 0.50 |
| MAML + Meta-dropout [35] | ICLR-20 | 51.93 ± 0.67% | 67.42 ± 0.52 |
| Meta-SGD + Meta-dropout [35] | ICLR-20 | 50.87 ± 0.63% | 65.55 ± 0.57 |
| MAML + L2F [36] | CVPR-20 | 52.10 ± 0.50% | 69.38 ± 0.46 |
| **AML** | / | **52.25 ± 0.85%** | **69.46 ± 0.68%** |
| SNAIL[14] | ICLR-18 | 55.71 ± 0.99% | 68.88 ± 0.92% |
| TADAM[64] | NIPS-18 | 58.50 ± 0.30% | 76.70 ± 0.30% |
| MetaGAN+RN[17] | NIPS-18 | 52.71 ± 0.64% | 68.63 ± 0.67% |
| AM3-TADAM[65] | ICLR-19 | **65.30 ± 0.49%** | 78.10 ± 0.36% |
| Incremental[66] | NIPS-19 | 54.95 ± 0.30% | 63.04 ± 0.30% |
| LEO+L2F[36] | CVPR-20 | 62.12 ± 0.13% | 78.13 ± 0.15% |
| **RAML** | / | 63.66 ± 0.85% | **80.49 ± 0.45%** |
| **URAML** | / | 49.56 ± 0.79% | 63.42 ± 0.76% |
| **URAML(AE)** | / | 33.29 ± 0.71% | 43.60 ± 0.66% |
| **URAML(RotNet)** | / | 53.61 ± 0.85% | 66.17 ± 0.73% |
| **URAML(MoCo)** | / | **55.73 ± 0.68%** | **69.32 ± 0.63%** |

#### 4.1.4. Other datasets

As the Representation module of URAML is trained by unsupervised learning, we take full advantage of this characteristic by training the Representation module of URAML on not only MiniImagenet-900 but also Places2 [56], COCO2017 [57], and OpenImages-300. The dataset OpenImages-300 is a subset of OpenImages-V4 [59]. The total OpenImages-V4 dataset contains 9 million images, and we randomly downloaded 3 million images from the OpenImages-V4 website to form the OpenImages-300 dataset.

### 4.2. Experiments on MiniImagenet

On MiniImagenet, we test all proposed methods on 5-way 1-shot and 5-way 5-shot classification tasks. All performances are presented in the $ACC_{mean} \pm ACC_{interval}$ formation, where $ACC_{mean}$ is the average of the testing accuracies on 600 testing tasks and $ACC_{interval}$ is the 95% confidence interval. All 600 testing tasks are randomly generated on the test set of MiniImagenet. The support and query set of each 5-way $K$-shot task contains $5*K$ and $5*15$ images, respectively.

**For AML**, the meta-leaner's network structure is shown in Fig. 2. The feature extractor contains 4 Convolution_ReLU_Batch normalization (C_R_B) layers and the classifier is a fully-connect layer. The attention model structure is shown in Fig. 3. Each Convolution layer consists of 64 channels. We train the meta-learner on 200,000 randomly generated tasks for 60,000 iterations, and set the learning rate to 0.001, and decay the learning rate to 0.0001 after 30,000 iterations. Moreover, to prevent the meta-learner from over-fitting, we set Dropout, L1, and L2 normalization to 0.2, 0.001, and 0.00001, respectively.

The experimental results of AML on MiniImagenet are shown in Table 1. Note that in Table 1, the methods shown in the upper part of the table use shallower backbone consisting of 4 or 5 C_R_B layers, while the methods in the lower part of the table use deeper ResNet-based backbone. Among all the methods using

**Table 2**
Detailed structure of the decoder in the auxiliary module of URAML.

| Layers | Number of filters | Kernel |
|---|---|---|
| Convolution | 1024 | 5 |
| Deconvolution | 512 | 3 |
| Deconvolution | 256 | 3 |
| Convolution | 1 or 2 | 1 |

shallow networks, AML attains the state-of-the-art performances on both the 5-way 1-shot and 5-way 5-shot image classification tasks. Note that

**For RAML**, the Representation module is a ResNet-50 [67] network, and the Auxiliary module is a fully-connect layer. The attention model is the same as that in AML, and the classifier contains two fully-connect layers.

At the prior-knowledge learning stage, we set the batch size and the learning rate to 256 and 0.001, respectively. L2 normalization with 0.00001 and Dropout with 0.2 are utilized to prevent the Representation module from over-fitting. At the meta-training stage, the ABP module is meta-trained with the same setting as AML. The experimental results of RAML are listed in Table 1. Compared to AML, RAML significantly improves the meta-learner's performance. It rises the accuracy on 5-way 1-shot tasks from 52.25% to 63.66%, and the accuracy on 5-way 5-shot tasks from 69.46% to 80.49%.

The most likely reason why RAML performs well is that before the meta-training stage, the Representation module has learned the prior-knowledge that helps the meta-learner understand new input images and provides high-level meaningful image representations. At the meta-training stage, the meta-learner's work becomes easier because it only needs learning how to quickly adjust its ABP module based on the compact features provided by the Representation module; it does not need to take care of the original high-dimensional input data. However, the AML meta-learner works harder than does the RAML meta-learner, because it must adjust its entire network to fit new few-shot learning tasks based on the original input data.

**For URAML**, the Representation module learns the prior-knowledge through unsupervised learning. As Fig. 5 shows, two independent ResNet-50 based encoders compose the Representation module. The Auxiliary module is composed of two deconvolution-based decoders. Table 2 shows the detail of the decoder network structure. The last Convolution layer's number of filters is set to 1 or 2 according to that the decoder recovers the $L$ channel or the $ab$ channels of the $Lab$ image. At both the prior-knowledge learning and meta-training stages, we set all hyperparameters to the same values as RAML. Note that to reduce the training computation cost, the decoders in the Auxiliary module recover the $ab$ and $L$ channels into $11 \times 11$ resolution rather than the original $84 \times 84$ resolution.

The experimental results of URAML are reported in Table 1. URAML lags behind RAML, and the underlying reason is that URAML learns prior-knowledge with unsupervised learning method Split-Brain, while RAML learns its prior-knowledge through supervised learning. The discussion about URAML(AE), URAML(RotNet), and URAML(MoCo) will be detailed in the ablation study in Section 4.4.2.

### 4.3. Experiments on Omniglot

As Omniglot is a much easier dataset than MiniImagenet that the existing meta-learners can easily achieve more than 95% accuracy on testing tasks generated on Omniglot, we only test method AML on Omniglot.

Same to the experiments on Miniimagenet, we also train the meta-learner on 200,000 randomly generated tasks for 60,000 iterations and set the learning rate to 0.001. The experiment results are shown in Table 3. It is clear that the proposed method AML attains state-of-the-art performance on 2 of all 4 kinds of few-shot image classification tasks. On the 5-way 1-shot task, though the method MetaGAN+RN slightly outperforms AML, we still highlight AML since MetaGAN+RN uses a deeper ResNet-based network while AML uses a shallower network. On the 20-way 1-shot task, our method AML surpasses other methods by a large margin. For example, compared to NIL, AML improves the meta-learner's performance from 96.70% to 98.48%.

### 4.4. Ablation study

#### 4.4.1. The attention mechanism

To confirm the beneficial effect of the attention mechanism for meta-learning, we conduct experiments to compare the performance of a meta-learner equipped with the attention model to that of its counterpart without the attention module. The experimental results are shown in Table 4. The compared meta-learner marked with an asterisk ('*') is the meta-learner re-implemented by ourselves. The performances of our re-implemented meta-learners differ slightly from those reported in their original papers. This is probably caused by different hyper-parameter or experimental settings (all methods in this experiment use convolutional layers with 32 filters). The comparisons in Table 4 reveals that the attention mechanism significantly improves the meta-learner.

#### 4.4.2. Unsupervised learning for URAML

URAML default learns prior-knowledge through Split-Brain. To investigate whether URAML is compatible with other unsupervised learning methods, we train URAML to learn prior-knowledge through other unsupervised learning methods Auto-Encoder [26], RotNet [46], and MoCo [32]. We use URAML (AE), URAML(RotNet) and URAML(MoCo) to denote the URAML meta-learners that learn prior-knowledge through Auto-Encoder, RotNet, and MoCo, respectively. The reported results of URAML (AE), URAML(RotNet), URAML(MoCo) and URAML in Table 1 indicate that the unsupervised learning algorithm selection substantially affects the meta-learner's performance. Clearly, URAML (MoCo) outperforms the other URAML meta-learners. The possible reason behind this result is that among all unsupervised learning methods, MoCo is the most powerful one. This experiment reveals that (1) URAML is compatible with other unsupervised learning methods and (2) better unsupervised learning methods result in better URAML performances. Considering that with the development of unsupervised learning algorithm, URAML's performance can be significantly improved, we also highlight the best performances of URAML.

#### 4.4.3. Prior-knowledge learning dataset

Here, we assess how does the prior-knowledge learning dataset affects RAML and URAML.

*(a) Effect to RAML*: In RAML, we adopted the reorganized *MiniImagenet-900* dataset as the default prior-knowledge learning dataset; however, in this experiment, the Representation module learns prior-knowledge from Places2 [56] instead. We denote this meta-learner as RAML(Places2). All the other experimental settings and hyperparameters are consistent with those conducted on the original RAML. Table 5 shows the experimental results. Clearly, the choice of prior-knowledge learning dataset affects the meta-learner. The reason is that different prior-knowledge learning datasets lead the Representation module to learn different knowledge and express the image features differently. Places2 is a dataset commonly used for scene classification that results in the

**Table 3**
Few-shot learning performance on Omniglot. The accuracy is assessed in the same way as MAML [10].

| Method | Venue | 5-way accuracy | | 20-way accuracy | |
|---|---|---|---|---|---|
| | | 1-shot | 5-shot | 1-shot | 5-shot |
| MAML [10] | ICML-17 | 98.70 ± 0.40% | 99.90 ± 0.10% | 95.80 ± 0.30% | 98.90 ± 0.20% |
| Prototypical nets [5] | NIPS-17 | 98.80% | 99.70% | 96.00% | 98.90% |
| Meta-SGD [11] | / | 99.53 ± 0.26% | **99.93 ± 0.09%** | 95.93 ± 0.38% | 98.97 ± 0.19% |
| Relation net [60] | CVPR-18 | 99.60 ± 0.20% | 99.80 ± 0.10% | 97.60 ± 0.20% | 99.10 ± 0.10% |
| GNN [61] | ICLR-18 | 99.20% | 99.70% | 97.40% | 99.00% |
| Spot-Learn [62] | CVPR-19 | 97.56 ± 0.31% | 99.65 ± 0.06% | / | / |
| iMAML HF [33] | NIPS-19 | 99.50 ± 0.26% | 99.74 ± 0.11% | 96.18 ± 0.36% | 99.14 ± 0.10% |
| SNAIL[14] | ICLR-18 | 99.07 ± 0.16% | 99.78 ± 0.09% | 97.64 ± 0.30% | 99.36 ± 0.18% |
| NIL [39] | ICLR-20 | / | / | 96.70 ± 0.30% | 98.00 ± 0.04% |
| MetaGAN + RN[17] | NIPS-18 | **99.67 ± 0.18%** | 99.86 ± 0.11% | 97.64 ± 0.17% | 99.21 ± 0.10% |
| **AML(ours)** | / | **99.65 ± 0.10%** | 99.85 ± 0.04% | **98.48 ± 0.09%** | **99.55 ± 0.06%** |

**Table 4**
Ablation experiments to investigate the attention mechanism on MiniImagenet.

| Method | 5-way accuracy | |
|---|---|---|
| | 1-shot | 5-shot |
| MAML* | 48.03 ± 0.83% | 64.11 ± 0.73% |
| MAML+attention | **48.52 ± 0.85%** | **64.94 ± 0.69%** |
| Reptile* | 48.23 ± 0.43% | 63.69 ± 0.49% |
| Reptile+attention | **48.30 ± 0.45%** | **64.22 ± 0.39%** |
| Meta-SGD* | 48.15 ± 0.93% | 63.73 ± 0.85% |
| Meta-SGD+attention | **49.11 ± 0.94%** | **65.54 ± 0.84%** |

Representation module learning scene understanding knowledge rather than object classification knowledge.

*(b) Effect to URAML*: In this experiment, we test how the quantity of unlabeled *Lab* images in the prior-knowledge learning dataset affect URAML. We train two other versions of URAML: URAML(V1) and URAML(V2). The Representation module of URAML(V1) learns prior-knowledge only from MiniImagenet-900, while that of URAML(V2) learns prior-knowledge from three datasets: MiniImagenet-900, Places2, and COCO2017. Compared with URAML(V1) and URAML(V2), the primordial URAML uses the largest quantity of unlabeled *Lab* images because it learns on all of MiniImagenet-900, Places2, COCO2017, and OpenImages-300. Table 5 shows the performances of URAML(V1) and URAML(V2). Clearly, the primordial URAML outperforms URAML(V1) and URAML(V2), and the more unlabeled *Lab* images that the meta-learner uses to learn prior-knowledge the better it performs. This experiment reveals that a large performance improvement space still exists, because we can use more unlabeled data to train URAML.

### 4.5. Cross-testing experiment

We find that the existing meta-learning methods generally suffer from a seldom-studied problem: Task-Over-Fitting (TOF). An example of the TOF problem is that a meta-learner to be tested on 5-way 1-shot classification tasks should be trained on 5-way 1-shot tasks rather than on other tasks. Similarly, a meta-learner to be tested on 5-way 5-shot tasks should be trained on 5-way 5-shot tasks. This situation occurs because a meta-learner trained on 5-shot tasks overfits to 5-shot tasks, and when tested on 1-shot tasks, it will perform notably worse than the meta-learner trained on 1-shot tasks.

We perform numerous cross-testing experiments to evaluate the extent to which MAML, Meta-SGD, AML, RAML, and URAML suffer from the TOF problem. The corresponding experimental results show that compared with the other methods, the proposed methods suffer less from the TOF problem, especially RAML and URAML.

For each tested meta-learning method, we perform cross-testing experiments in the following manner: (1) Train the meta-learner on 5-way $K$-shot image classification tasks, where $K \in \{1, 3, 5, 7, 9\}$. (2) Test the meta-learner on 5-way $J$-shot tasks, where $J \in \{1, 3, 5, 7, 9\}$. Finally, we can obtain $5 * 5 = 25$ cross-testing performance for each meta-learning method. The experimental results are shown in Table 6.

Table 6 reveals that MAML suffers extensively from the TOF problem because its meta-learner, which performs best on $K$-shot tasks, probably does not perform well on $J$-shot tasks, where $K \neq J$. For example, on 1-shot testing tasks, the MAML meta-learner trained on 1-shot tasks performs best; but on 3-, 5-, 7-, and 9-shot testing tasks, it performs worse than the other four meta-learners trained on 3-, 5-, 7-, and 9-shot training tasks. This experiment means the MAML meta-learner trained on 1-shot training tasks overfits to 1-shot testing tasks. The URAML meta-learner is little troubled by the TOF problem because the meta-learner that performs best on $K$-shot tasks probably also performs best on $J$-shot tasks, where $K, J \in \{1,5,7,9\}$. For example, the URAML meta-learner trained on 1-shot training tasks performs best not only on 1-shot testing tasks but also on 5-, 7-, and 9-shot testing tasks, which means that the meta-learner trained on 1-shot tasks generalizes well to the other $J$-shot testing tasks.

We design a Cross-Entropy across Tasks (CET) metric to quantize the extent to which a given meta-learning approach is vulnerable to the TOF problem. The evaluation process is shown in Eq. (14), where $i, j \in \{1,3,5,7,9\}$. The overstruck variables are vectors, and $\mathcal{S}$ and $\mathcal{D}$ are the softmax and cross-entropy operations, respectively. $\boldsymbol{a}_i$ represents the $i$-shot testing accuracies ($i$-shot row in Table 7) of the five meta-learners trained on 1-, 3-, 5-, 7-, and 9-shot tasks. $\boldsymbol{d}_i$ represents the $i$-shot testing accuracy distribution of the meta-learners. $l_{i,j}$ represents the similarity between the accuracy distribution vector $\boldsymbol{d}_i$ and $\boldsymbol{d}_j$, where $i,j \in \{1,3,5,7,9\}$. $L$ represents the overall similarities of $l_{i,j}$ for a specific approach.

$$\begin{cases} \boldsymbol{d}_i = \mathcal{S}(\boldsymbol{a}_i / max(\boldsymbol{a}_i)) \\ l_{ij} = \mathcal{D}(\boldsymbol{d}_i, \boldsymbol{d}_j) \\ L = \sum_{i,j \in 1,3,5,7,9}^{i \neq j} l_{ij} \end{cases} \tag{14}$$

For example, the testing accuracies $\boldsymbol{a}_3$ of Meta-SGD [58.24%, 59.18%, 58.90%, 58.75%, 59.15%] are the 3-shot testing accuracies of the five Meta-SGD meta-learners trained on 1-, 3-, 5-, 7-, and 9-shot tasks. Therefore, $\boldsymbol{a}_3 / max(\boldsymbol{a}_3)$ = [58.24%, 59.18%, 58.90%, 58.75%, 59.15%] / 59.18%, and $\boldsymbol{d}_3 = \mathcal{S}(\boldsymbol{a}_3 / max(\boldsymbol{a}_3))$ = [0.116, 0.255, 0.202, 0.178, 0.249]. Similarly, $\boldsymbol{d}_7$ = [0.122, 0.206, 0.255, 0.233, 0.184]. Then, $l_{3,7} = 1.603$ and $L = 34.22$.

Obviously, the smaller the total distance $L$ appears, the less the meta-learning approach suffers from the TOF problem. We show the performances of the different meta-learning approaches in terms of the CET metric in Table 7. This experiment shows that the proposed AML, RAML, and URAML perform better than do MAML and Meta-SGD on the CET metric; of these, RAML and URAML perform the best. The possible reason for this result is that prior-knowledge and attention mechanism are both helpful

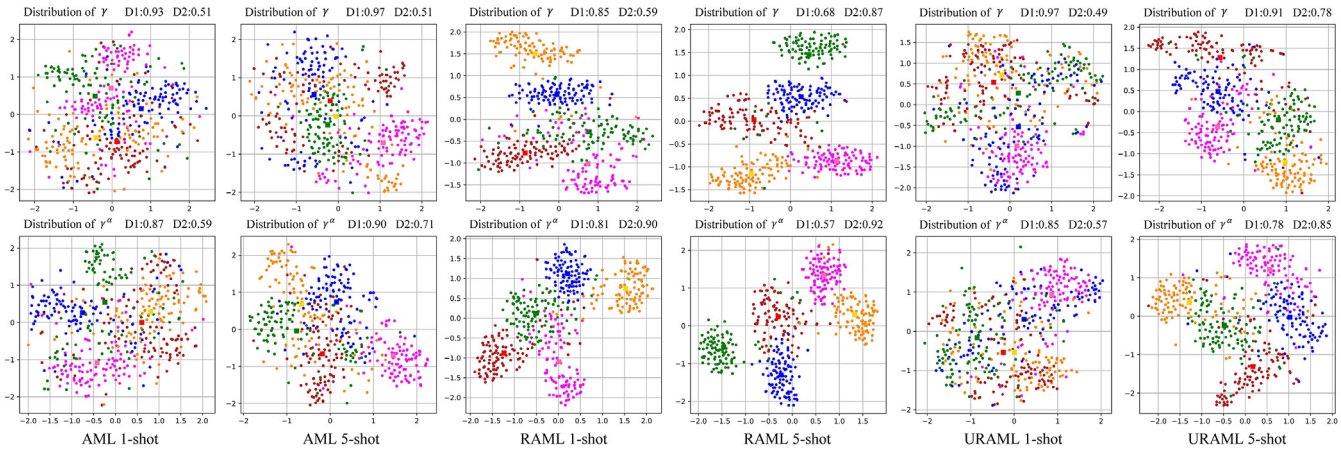


**Fig. 6.** Visualization of the distributions of the $\gamma$ and $\gamma^{\alpha}$ features of all three developed methods. For each method, we randomly generate a 5-way 1-shot and a 5-way 5-shot testing task on Miniimagenet. The query set in each task contains 100 images for each image class. For each testing task, after the meta-learner inner-updating on the support set, we use t-SNE to visualize the distributions of the meta-learner's $\gamma$ and $\gamma^{\alpha}$ of the query set images. In each picture, five colors are used to represent the 5 image classes in the testing task and each point denotes the $\gamma$ or $\gamma^{\alpha}$ feature of a query image. We also show the inner-class distance D1 and inter-class distance D2 of the feature distribution in each picture to better understand the distributions.

**Table 5**
Results of the ablation experiments evaluating the effects of prior-knowledge learning dataset to RAML and URAML. P: Places2, M: Miniimagenet-900, C: COCO2017, O: Openimages-300.

| Method | Dataset | Number of images | 5-way accuracy | |
|---|---|---|---|---|
| | | | 1-shot | 5-shot |
| RAML(Places2) | P | 2.62 million | 58.82 ± 0.89% | 74.09 ± 0.76% |
| RAML | M | 1.15 million | **63.66 ± 0.85%** | **80.49 ± 0.45%** |
| URAML(V1) | M | 1.15 million | 45.91 ± 0.79% | 61.04 ± 0.71% |
| URAML(V2) | M, P, C | 4.10 million | 48.82 ± 0.79% | 62.84 ± 0.78% |
| URAML | M, P, C, O | 7.10 million | 49.56 ± 0.79% | 63.42 ± 0.76% |

**Table 6**
Cross-testing experimental results of MAML, Meta-SGD, AML, RAML, and URAML. Each column presents specific $K$-shot training tasks and each row presents specific $J$-shot testing tasks, where $K, J \in \{1,3,5,7,9\}$. For each method, the value in the $J$-shot row and $K$-shot column presents the $J$-shot testing accuracy of the meta-learner trained on $K$-shot training tasks. For example, the value 59.99 in the 3-shot row and 7-shot column of MAML presents the 3-shot testing accuracy of the MAML meta-learner trained on 7-shot training tasks. The value 80.83 in the 9-shot row and 1-shot column of RAML presents the 9-shot testing accuracy of the RAML meta-learner trained on 1-shot training tasks. For each method, within each $J$-shot testing scene, we highlight the highest accuracy with bold font. Moreover, the higher the accuracy the more red background. The worst accuracy on each $J$-shot testing scene uses white background.

| Method | Meta-SGD | | | | | MAML | | | | | AML | | | | | RAML | | | | | URAML | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Testing task \ Training task | 1-shot | 3-shot | 5-shot | 7-shot | 9-shot | 1-shot | 3-shot | 5-shot | 7-shot | 9-shot | 1-shot | 3-shot | 5-shot | 7-shot | 9-shot | 1-shot | 3-shot | 5-shot | 7-shot | 9-shot | 1-shot | 3-shot | 5-shot | 7-shot | 9-shot |
| 1-shot | **48.15** | 47.71 | 46.44 | 46.67 | 46.38 | **48.03** | 45.56 | 40.54 | 41.07 | 39.69 | **52.25** | 51.58 | 51.79 | 51.66 | 51.03 | **63.66** | 63.54 | 63.14 | 63.46 | 63.48 | **49.56** | 48.04 | 47.89 | 47.43 | 46.26 |
| 3-shot | 58.24 | **59.18** | 58.90 | 58.75 | 59.15 | 58.49 | **60.12** | 59.12 | 59.99 | 59.43 | 63.23 | 64.97 | **65.19** | 65.02 | 64.56 | 74.73 | 76.60 | 76.18 | 76.58 | **76.78** | 59.03 | 58.48 | 58.90 | **59.32** | 58.21 |
| 5-shot | 62.85 | 63.56 | 63.73 | **63.93** | 63.79 | 62.06 | 64.18 | 64.11 | **64.32** | 64.31 | 67.64 | 68.82 | **69.46** | 69.32 | 69.08 | 78.10 | 80.15 | **80.49** | 80.17 | 80.29 | **63.64** | 62.96 | 63.42 | 62.04 | 62.22 |
| 7-shot | 65.10 | 65.79 | **66.07** | 65.95 | 65.64 | 64.52 | 66.85 | **67.42** | 67.33 | 67.19 | 69.51 | 71.35 | 71.60 | 72.57 | **72.68** | 79.93 | 82.19 | **82.62** | 82.28 | 82.31 | **65.52** | 64.54 | 63.77 | 64.60 | 64.06 |
| 9-shot | 66.25 | 67.04 | 67.36 | 67.53 | **67.73** | 65.06 | 68.44 | 69.01 | **69.49** | 69.31 | 71.32 | 73.16 | 73.43 | **73.76** | 73.12 | 80.83 | 83.69 | 83.46 | 83.51 | **83.72** | **66.82** | 66.37 | 65.87 | 65.85 | 65.30 |

**Table 7**
Performance of different meta-learning methods on the CET metric.

| Method | MAML | Meta-SGD | AML | RAML | URAML |
|---|---|---|---|---|---|
| CET | 57.19 | 34.22 | 33.35 | **32.13** | 32.16 |

in reducing the meta-learner's few-shot cognitive load and in avoiding interference from redundant useless information.

An interesting phenomenon can be seen in Table 6, where the meta-learner trained by RAML on 5-way 9-shot tasks performs best on most of the test tasks, but the meta-learner trained by URAML on 5-way 1-shot tasks performs best. The possible reason behind this phenomenon is that the Representation module of RAML learns knowledge through supervised learning, while the Representation module of URAML learns knowledge through unsupervised learning. This difference results in variations in the output features between these two types of Representation modules.

### 4.6. Feature distribution

To understand the effect of attention mechanism, we visualize the distributions of features $\gamma$ and $\gamma^{\alpha}$ (shown in Figs. 2, 4 and 5) in Fig. 6 with t-SNE [68]. In Fig. 6, the 500 feature points of each picture represent 500 $\gamma$ or $\gamma^{\alpha}$ features from the query set images of a randomly generated 5-way 1- or 5-shot testing task on MiniImagenet.

The average distribution inner-class distance D1 of $\gamma^{\alpha}$ is smaller than that of $\gamma$, and the average inter-class distance D2 of $\gamma^{\alpha}$ is larger than that of $\gamma$. This result indicates that among different image classes, the distribution of $\gamma^{\alpha}$ is more distinguishable than that of $\gamma$. The underlying reason is that the attention mechanism enables the meta-learner to quickly focus its attention on the critical image features and makes $\gamma^{\alpha}$ more distinguishable than $\gamma$ for differentiating images of different classes.

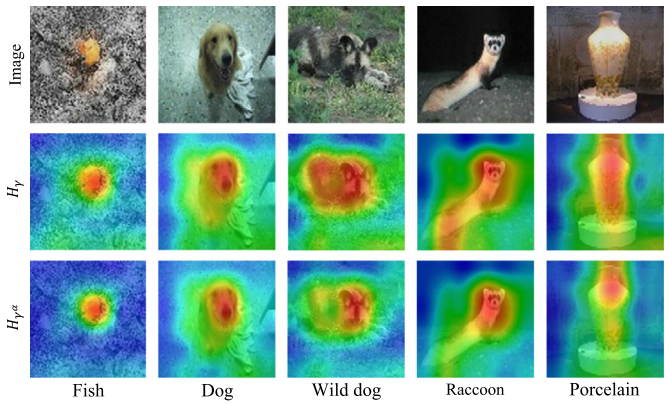


**Fig. 7.** The first row shows some images sampled from the query set of a 5-way 1-shot classification task. The second and third rows show the corresponding $H_\gamma$ and $H_{\gamma^\alpha}$ heat-maps.

### 4.7. Heat-maps

To further analyze how the meta-learner leverages its attention model, we visualize the $\gamma$ and $\gamma^\alpha$ heat-maps in Fig. 7, where the heat-maps are denoted as $H_\gamma$ and $H_{\gamma^\alpha}$, respectively. $\gamma$ is the output of RAML's Representation module, while $\gamma^\alpha$ is the attentive feature. To obtain $H_\gamma$, we first inner-update the RAML meta-learner on the support set of a randomly generated 5-way 1-shot testing task on MiniImagenet. Second, we input each image $x \in \mathbb{R}^{84\times84\times3}$ of the query set to the meta-learner and obtain the feature map $\gamma \in \mathbb{R}^{6\times6\times2048}$. Third, we average $\gamma$ across its 2048 channels to obtain a feature map $\mathcal{M}_\gamma \in \mathbb{R}^{6\times6}$. Then, we convert the gray feature map $\mathcal{M}_\gamma$ into an RGB heat-map $\hat{\mathcal{M}}_\gamma \in \mathbb{R}^{6\times6\times3}$. Finally, we resize $\hat{\mathcal{M}}_\gamma$ into $84 \times 84$ resolution and obtain $H_\gamma \in \mathbb{R}^{84\times84\times3}$ with the formulation $H_\gamma = 0.4 * x + 0.6 * \hat{\mathcal{M}}_\gamma$. Similarly, we can obtain the heat-map $H_{\gamma^\alpha} \in \mathbb{R}^{84\times84\times3}$ through the same procedure.

The heat-maps in Fig. 7 show that $H_{\gamma^\alpha}$ is more sensitive to the distinguishable region of the input image than is $H_\gamma$, revealing that with the attention model, the meta-learner learns to focus on the most discriminative image features. For example, the first column of Fig. 7 is a fish. In addition to the fish body, $H_\gamma$ is also sensitive to some background regions of the image. However, the meta-learner discovers that the fish body is the only crucial feature for categorizing this image; thus, it shrinks its attention region so that $H_{\gamma^\alpha}$ is sensitive only to the fish body.

Through the visualization and analysis of the heat-maps $H_\gamma$ and $H_{\gamma^\alpha}$, we conclude that the attention mechanism helps the meta-learner focus on the most distinguishable image features, further helping the meta-learner perform better few-shot learning.

## 5. Conclusion and the future work

In this paper, inspired by human cognition and learning process, we investigate the importance of attention mechanism and prior-knowledge for meta-learning based few-shot learning. To solve a few-shot learning task, the meta-learner should first well use stable prior-knowledge to understand images and extract compact image feature representations, allowing it to solve the task in the compact feature representation space rather than the high-dimensional original image space. Then, the meta-learner should focus its attention on the crucial aspects of the extracted feature representations and make the final decision based on that attention. Therefore, we propose three step-by-step methods, AML, RAML, and URAML, to introduce attention mechanism and prior-knowledge to meta-learning. All three methods work successfully and achieve state-of-the-art performances on a variety of few-shot learning benchmarks, which indicates the rationality of our viewpoint and methods.

In addition, we find that the existing meta-learning approaches suffer from the TOF problem, making it difficult for these meta-learning approaches to be deployed in practical few-shot learning applications. We design a novel CET metric to evaluate the extent to which a meta-learning method suffers from TOF. The experiments show that compared to the existing meta-learning methods, the proposed methods suffer less from the TOF problem, particularly RAML and URAML.

Among all the proposed methods, although URAML does not perform best, we believe it is the most promising method because considerable development space exists for improving its performance. From our experiments, two manners seem to be able to significantly improve the performance of URAML. One manner involves developing better unsupervised or self-supervised learning algorithms. RAML performs better than URAML, revealing that the current unsupervised learning algorithm falls behind supervised learning. Bridging the gap between unsupervised learning and supervised learning algorithms would boost the performance of URAML by a substantial amount. The other manner involves using more unlabeled data from which URAML can learn prior-knowledge. Although our current approach uses 7.1 million unlabeled images to train URAML, that is still dramatically smaller than the number of images that humans have typically seen in terms of both quantity and quality. Regarding quantity, we can assume that if a person views 1 image per second continually for 15 h per day, that person will have seen about 100 million images in 5 years. Regarding quality, humans see the world in a multimodal manner; that is, humans can not only see objects but also touch them and move them around (or move around the objects); this aspect helps humans understand the world more accurately than can computer vision alone. In a word, developing unsupervised or self-supervised learning algorithms and collecting more unlabeled images would both help to improve URAML's performance.

## CRediT authorship contribution statement

**Yunxiao Qin:** Methodology, Software, Writing - original draft. **Weiguo Zhang:** Supervision, Funding acquisition. **Chenxu Zhao:** Investigation, Data curation. **Zezheng Wang:** Visualization. **Xiangyu Zhu:** Validation, Writing - original draft. **Jingping Shi:** Conceptualization. **Guojun Qi:** Formal analysis. **Zhen Lei:** Resources, Writing - review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This work is supported in part by the National Key Research and Development Program of China (No. 2020YFC2003901) and in part by the National Natural Science Foundation of China (No. 61573286, 61876178, and 61976229).

## References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[2] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning., in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, Vol. 4, 2017, pp. 4278–4284.
[3] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: International Conference on Learning Representations, 2014.
[4] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, 2016, pp. 3630–3638.
[5] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4077–4087.
[6] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, J. Luo, Revisiting local descriptor based image-to-class measure for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7260–7268.
[7] Y. Bengio, S. Bengio, J. Cloutier, Learning a Synaptic Learning Rule, Université de Montréal, Département d'informatique et de recherche opérationnelle, 1990.
[8] S. Bengio, Y. Bengio, J. Cloutier, J. Gecsei, On the optimization of a synaptic learning rule, in: Preprints Conf. Optimality in Artificial and Biological Neural Networks, Univ. of Texas, 1992, pp. 6–8.
[9] J. Schmidhuber, Learning to control fast-weight memories: An alternative to dynamic recurrent networks, Neural Comput. 4 (1) (1992) 131–139.
[10] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, 2017, arXiv:1703.03400.
[11] Z. Li, F. Zhou, F. Chen, H. Li, Meta-sgd: Learning to learn quickly for few shot learning, 2017, arXiv preprint arXiv:1707.09835.
[12] S. Ravi, H. Larochelle, Optimization as a model for few-shot learning, in: International Conference on Learning Representations, 2017.
[13] A. Nichol, J. Achiam, J. Schulman, On first-order meta-learning algorithms, 2018, arXiv preprint arXiv:1803.02999.
[14] N. Mishra, M. Rohaninejad, X. Chen, P. Abbeel, A simple neural attentive meta-learner, in: International Conference on Learning Representations, 2018.
[15] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367–4375.
[16] Y. Duan, J. Schulman, X. Chen, P.L. Bartlett, I. Sutskever, P. Abbeel, $RL^2$: Fast reinforcement learning via slow reinforcement learning, 2016, arXiv preprint arXiv:1611.02779.
[17] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, Y. Song, Metagan: An adversarial approach to few-shot learning, in: Advances in Neural Information Processing Systems, 2018, pp. 2367–2376.
[18] T. Munkhdalai, H. Yu, Meta networks, Proc. Mach. Learn. Res. 70 (2017) 2554.
[19] J.A. Langer, M. Nicolich, Prior knowledge and its relationship to comprehension, J. Read. Behav. 13 (4) (1981) 373–379.
[20] F. Dochy, Instructional implications of recent research and empirically-based theories on the effect of prior knowledge on learning, in: Learning Environments, Springer, 1990, pp. 339–355.
[21] A.M. Shapiro, How including prior knowledge as a subject variable may change outcomes of learning research, Amer. Educ. Res. J. 41 (1) (2004) 159–189.
[22] J. Wylie, C. McGuinness, The interactive effects of prior knowledge and text structure on memory for cognitive psychology texts, British J. Educ. Psychol. 74 (4) (2004) 497–514.
[23] I. Hsin, F. Paas, Effects of computer-based visual representation on mathematics learning and cognitive load, J. Educ. Technol. Soc. 18 (4) (2015) 70–77.
[24] G. Logan, D. Dagenbach, T. Carr, Inhibitory Processes in Attention, Memory and Language, Academic Press, San Diego, 1994, pp. 189–239.
[25] S.A. Hillyard, E.K. Vogel, S.J. Luck, Sensory gain control (amplification) as a mechanism of selective attention: electrophysiological and neuroimaging evidence, Philos. Trans. R. Soc. B 353 (1373) (1998) 1257–1270.
[26] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[27] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.
[28] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.
[29] R. Zhang, P. Isola, A.A. Efros, Colorful image colorization, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2016, pp. 649–666.
[30] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 132–149.
[31] R. Zhang, P. Isola, A.A. Efros, Split-brain autoencoders: Unsupervised learning by cross-channel prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1058–1067.
[32] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.
[33] A. Rajeswaran, C. Finn, S.M. Kakade, S. Levine, Meta-learning with implicit gradients, in: Advances in Neural Information Processing Systems, 2019, pp. 113–124.
[34] E. Grant, C. Finn, S. Levine, T. Darrell, T. Griffiths, Recasting gradient-based meta-learning as hierarchical bayes, in: International Conference on Learning Representations, 2018.
[35] B.H. Lee, T. Nam, E. Yang, J.S. Hwang, Meta dropout: Learning to perturb latent features for generalization, in: International Conference on Learning Representations, 2020.
[36] S. Baik, S. Hong, M.K. Lee, Learning to forget for meta-learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 2379–2387.
[37] S. Flennerhag, A.A. Rusu, R. Pascanu, F. Visin, H. Yin, R. Hadsell, Meta-learning with warped gradient descent, in: International Conference on Learning Representations, 2020.
[38] A. Antoniou, H. Edwards, J.A. Storkey, How to train your maml, in: International Conference on Learning Representations, 2019.
[39] A. Raghu, M. Raghu, S. Bengio, O. Vinyals, Rapid learning or feature reuse? Towards understanding the effectiveness of MAML, in: International Conference on Learning Representations, 2020.
[40] Q. Sun, Y. Liu, T. Chua, B. Schiele, Meta-transfer learning for few-shot learning, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, pp. 403–412.
[41] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3156–3164.
[42] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
[43] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, in: Advances in Neural Information Processing Systems, 2014, pp. 2204–2212.
[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[45] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[46] S. Gidaris, P. Singh, N. Komodakis, Unsupervised representation learning by predicting image rotations, in: International Conference on Learning Representations, 2018.
[47] Z. Feng, C. Xu, D. Tao, Self-supervised representation learning by rotation feature decoupling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10364–10374.
[48] C. Ting, K. Simon, N. Mohammad, H. Geoffrey, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning 2020, 2020.
[49] T. Ormerod, Human cognition and programming, in: Psychology of Programming, Elsevier, 1990, pp. 63–82.
[50] A. Oliva, A. Torralba, M.S. Castelhano, J.M. Henderson, Top-down control of visual attention in object detection, in: Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429), Vol. 1, IEEE, 2003, pp. I–253.
[51] A.H. van der Heijden, Selective Attention in Vision, Routledge, 2003.
[52] M.I. Posner, S.E. Petersen, The attention system of the human brain, Annu. Rev. Neurosci. 13 (1) (1990) 25–42.
[53] S.K. Ungerleider, G. Leslie, Mechanisms of visual attention in the human cortex, Annu. Rev. Neurosci. 23 (1) (2000) 315–341.
[54] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, Z. Wang, Is the deconvolution layer the same as a convolutional layer?, 2016, arXiv preprint arXiv:1609.07009.
[55] B. Lake, R. Salakhutdinov, J. Gross, J. Tenenbaum, One shot learning of simple visual concepts, in: Proceedings of the Annual Meeting of the Cognitive Science Society, Vol. 33, 2011.
[56] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. (2017).

[57] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, Ieee, 2009, pp. 248–255.

[59] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, S. Kamali, M. Malloci, J. Pont-Tuset, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, K. Murphy, Openimages: A public dataset for large-scale multi-label and multi-class image classification, 2017, Dataset available from https://storage.googleapis.com/openimages/web/index.html.

[60] F.S.Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: Relation network for few-shot learning, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018.

[61] V. Garcia, J. Bruna, Few-shot learning with graph neural networks, in: International Conference on Learning Representations, 2017.

[62] W.-H. Chu, Y.-J. Li, J.-C. Chang, Y.-C.F. Wang, Spot and learn: A maximum-entropy patch sampler for few-shot image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6251–6260.

[63] P. Zhou, X. Yuan, H. Xu, S. Yan, J. Feng, Efficient meta learning via minibatch proximal update, in: Advances in Neural Information Processing Systems, 2019, pp. 1532–1542.

[64] B.N. Oreshkin, A. Lacoste, P. Rodriguez, Tadam: Task dependent adaptive metric for improved few-shot learning, in: Advances in Neural Information Processing Systems, 31, 2018, pp. 721–731.

[65] C. Xing, N. Rostamzadeh, B.N. Oreshkin, P.O. Pinheiro, Adaptive cross-modal few-shot learning, in: Advances in Neural Information Processing Systems, 32, 2019, pp. 4847–4857.

[66] M. Ren, R. Liao, E. Fetaya, R. Zemel, Incremental few-shot learning with attention attractor networks, in: Advances in Neural Information Processing Systems, 2019, pp. 5276–5286.

[67] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[68] L.v.d. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.