

Article

Task-Covariant Representations for Few-Shot Learning on Remote Sensing Images

Liyi Zhang, Zengguang Tian, Yi Tang * and Zuo Jiang *

School of Mathematics and Computer Science, Yunnan Minzu University, Kunming 650500, China; 22213037570005@ymu.edu.cn (L.Z.); 10219130407@ymu.edu.cn (Z.T.)

* Correspondence: yitang4math@ynni.edu.cn (Y.T.); jiangzuo@ymu.edu.cn (Z.J.)

Abstract: In the regression and classification of remotely sensed images through meta-learning, techniques exploit task-invariant information to quickly adapt to new tasks with fewer gradient updates. Despite its usefulness, task-invariant information alone may not effectively capture task-specific knowledge, leading to reduced model performance on new tasks. As a result, the concept of task-covariance has gained significant attention from researchers. We propose task-covariant representations for few-shot Learning on remote sensing images that utilizes capsule networks to effectively represent the covariance relationships among objects. This approach is motivated by the superior ability of capsule networks to capture such relationships. To capture and leverage the covariance relations between tasks, we employ vector capsules and adapt our model parameters based on the newly learned task covariance relations. Our proposed meta-learning algorithm offers a novel approach to effectively address the real task distribution by incorporating both general and specific task information. Based on the experimental results, our proposed meta-learning algorithm shows a significant improvement in both the average accuracy and training efficiency compared to the best model in the experiments. On average, the algorithm increases the accuracy by approximately 4% and improves the training efficiency by approximately 8%.

Keywords: capsule network; meta-learning; subspace learning; covariance

MSC: 68T45



Citation: Zhang, L.; Tian, Z.; Tang, Y.; Jiang, Z. Task-Covariant Representations for Few-Shot Learning on Remote Sensing Images. *Mathematics* **2023**, *11*, 1930. <https://doi.org/10.3390/math11081930>

Academic Editors: Xiangtao Zheng, Jinchang Ren and Ling Wang

Received: 15 March 2023
Revised: 13 April 2023
Accepted: 14 April 2023
Published: 19 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Task-invariant representation meta-learning methods [1–3] apply prior knowledge learned from previous tasks to new ones and has been successfully used in few-shot learning problems, such as classification and regression. However, the prerequisite for the success of these methods is the invariant representation of all tasks, with no differences among tasks, and good knowledge transfer can be achieved by learning globally shared meta-knowledge for all tasks. The form of meta-knowledge includes the model structure, initialization parameters, and loss function. Most existing meta-learning methods that learn globally shared meta-knowledge fail to handle all tasks, because it is necessary to consider the covariance among tasks, i.e., that there are both differences and correlations among tasks. According to the covariant relationship between tasks, the learned meta-knowledge is adaptively adjusted for each new task, which can address the deficiency of task-invariant representation meta-learning, in which all tasks utilize the learned globally shared meta-knowledge.

The covariant relationship among tasks should consider not only the differences among tasks but also the correlations among tasks. The factors that affect the learning effect of each task include the initialization parameters, loss function, and update rules. To address the problem of invariant representation meta-learning, where all tasks utilize the learned globally shared meta-knowledge, the concept of heterogeneity among tasks has been proposed [4–6], which is based on differences among tasks and uses task-specific

information to customize specific meta-knowledge for each task. Several works have attempted to solve this problem by attempting to find better initialization parameters [7–10]. Baik et al. proposed a method to adaptively learn the loss function for each task during the inner-loop optimization of meta-learning [11]. Several other works have developed faster adaptation processes, for example, by increasing the learning rate [12,13]. Considering the importance of tasks, some studies have achieved better results by weighting different tasks in the outer-loop update [14,15]. However, in addressing the problem of learning globally shared meta-knowledge for all tasks, the above methods only consider differences among tasks, and correlations among tasks are not taken into account.

A capsule is a group of neurons whose activity vector represents the instantiation parameters of a specific type of entity, such as an object or an object part [16–19]. In contrast to convolutional neural networks, it not only represents the presence of objects but also more prominently represents the covariance between objects. Covariant relationships can effectively describe the subtle differences among objects of the same category and enhance the expressiveness of objects. Inspired by the covariant representation capability of capsule networks, we propose a task-covariant representation meta-learning algorithm. The real task distribution may contain several subdistributions with large differences, and, using the covariant relationship among tasks, different subspace representations are learned for different subdistributions. The correlations among tasks are reflected within subspaces, and the differences between tasks are reflected among the subspaces. Different modulation functions are learned for different subspaces, and the learned meta-knowledge is modulated by using the task information and corresponding modulation functions in the subspaces so as to achieve the fast learning and generalization of new tasks, as shown in the covariance subplot in Figure 1.

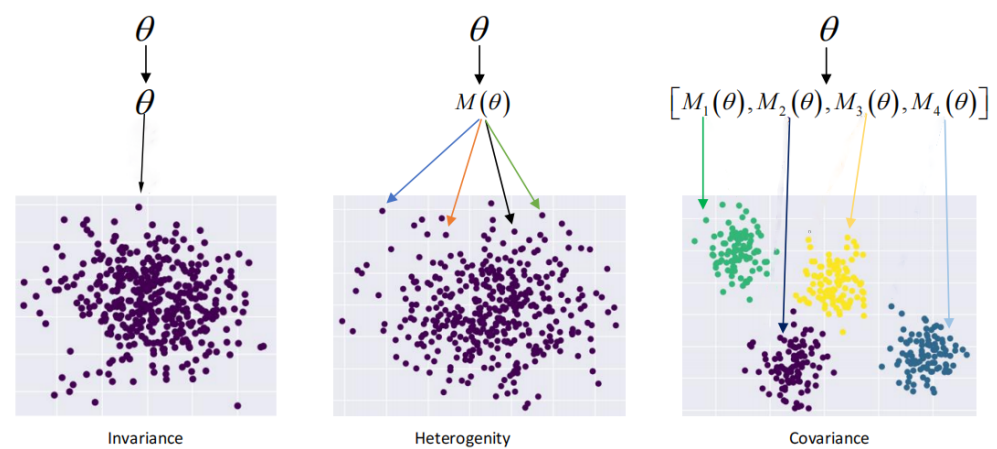


Figure 1. Task-invariant, task-heterogeneous, and task-covariant representation meta-learning comparison diagram. Task-invariant representation meta-learning learns globally shared meta-knowledge for all tasks, task-heterogeneous meta-learning uses task information to adaptively modulate learned meta-knowledge, and task-covariant representation means that meta-learning uses the covariant relationships among tasks to assign the current task to the corresponding subspace and uses the modulation function and task information corresponding to the subspace to adaptively modulate the learned meta-knowledge.

The key challenges in meta-learning of task-covariant representations are how current tasks are assigned to the corresponding subspaces and how to learn the corresponding modulation functions for each subspace. In the task-covariant representation meta-learning algorithm, capsule representation is used to represent a task feature or a part of the task feature, and the dynamic routing algorithm [16] is used to represent the covariant relationships among tasks. In the dynamic routing algorithm, the number of output capsules in the final layer indicates the number of subspaces. Each output capsule's module length is computed and compared to find the capsule with the highest probability of existence,

which corresponds to the chosen subspace capsule. All other subspace capsules are then set to zero using masking. The subspace capsule is converted into a vector format and then used as input for the modulation function of the fully connected network. According to the different positions of the retained task capsules in the vector and the characteristics of the fully connected function, the corresponding modulation function can be learned for each subspace capsule, and the learned meta-knowledge can be modulated in combination with the current task information so as to obtain a solution more suitable for each task's meta-knowledge.

The proposed task-covariant representation meta-learning algorithm uses the capsule network to accurately express the relationships among task covariates. By assigning corresponding capsule subspaces to different subdistribution tasks and learning the modulation function corresponding to each subspace, the learned meta-knowledge is better modulated using the current task information and the corresponding modulation function. Experimental results show that the proposed novel meta-learning algorithm can obtain results comparable to advanced algorithms in terms of accuracy and can effectively increase the model training performance.

The major contributions of our proposed task-covariant representation meta-learning method are summarized as follows:

- Considering the covariant relationships among tasks in practical use, a task-covariant representation meta-learning algorithm is proposed.
- According to the different distributions of tasks, the corresponding subspaces are allocated to different subdistributions.
- A corresponding modulation function is learned for each subspace, and the learned meta-knowledge is adaptively adjusted according to the task information and the corresponding modulation function.

2. Related Work

2.1. Meta-Learning

“Meta-learning” or “learning to learn” [20,21] is the process of distilling learning experience across multiple distribution-related tasks and using the extracted experience to improve future learning performance. Meta-learning aims to provide an integrated combination of next-step features, models, and algorithms, with the goal of replacing hand-designed learners relying on prior knowledge with learned learning algorithms [20,22,23]. Model-agnostic meta-learning [1] adopts the invariant representation of tasks and can learn globally shared meta-knowledge for all tasks. When a new task arrives, it can quickly learn the globally shared meta-knowledge. However, globally shared meta-knowledge does not consider the differences among tasks [4,5] and cannot effectively handle tasks with different distributions. In order to address the limitation of globally shared meta-learning in adapting to different tasks, some studies have introduced the concept of task heterogeneity [4,5,24]. This approach improves the efficiency of learning in new tasks by adaptively adjusting the learned meta-knowledge for each task, which includes the initialization parameters, loss function, and update rules.

From the initialization parameter angle, some works have used probabilistic models to tailor the globally shared initialization to suit each task [2,25,26]. Yao et al. [9] proposed a method that constructs a meta-knowledge graph between tasks and adaptively adjusts the model initialization parameters through the relationship between the new task and the meta-knowledge graph so as to achieve the purpose of selecting the initialization parameters for each task. Baik et al. [27] proposed adaptive learning attenuation parameters to adaptively adjust the learned prior knowledge for each task. From the loss function angle, Baik et al. [11] proposed a method that learns a loss function that is adapted to each task based on the current task state during inner-loop optimization. From the update rule angle, Baik et al. [13] proposed a small meta-network that can adaptively generate per-step hyperparameters: learning rate and weight decay coefficients. From the perspective of task importance, inspired by curriculum learning [28,29] and hard sample learning [30,31], some

researchers have proposed hard task strategies. By resampling tasks with lower validation accuracy among the learned tasks, the difficult tasks are learned multiple times [32,33]; in model-agnostic meta-learning, according to the gradient value of each task in the inner loop and the similarity between the support set and the query set gradient, during the update of the outer loop, the weight of each task loss in the total loss function is assigned according to the different gradients [14]. From the perspective of task generation, the random sampling of tasks may be sub-optimal and uninformative, so adaptive sampling can be used to improve the performance and learning efficiency of the model. According to the impact of categories on learning efficiency, a greedy-based class-pair learning algorithm was proposed to generate the class-pair potential for difficult tasks [34]. For multi-modal tasks, Zheng et al. proposed a partially interactive collaboration method to reduce the modality gap in VI-ReID [35] by utilizing complementary information from different modalities. Yao et al. [5] used a hierarchical structure method that adaptively customizes the learned prior knowledge for each task set, but the prior knowledge for each task set is still customized on the basis of the globally shared prior knowledge. However, the above methods only consider the differences among tasks or task sets and do not consider the covariant relationships among tasks. We propose to adaptively adjust the learned meta-knowledge for each task according to the real distribution of tasks and the ability of the capsule network to represent the covariant relationships among tasks so as to improve the learning efficiency of the adjusted meta-knowledge for the current task. Many methods adopt a two-stream network and design additional constraint conditions to extract shared features for different modalities. However, the interaction between the feature extraction processes of different modalities is rarely considered. Finn et al. proposed Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks (MAML [1]). Li et al. proposed Learning to Learn Quickly for Few-Shot Learning (Meta-SGD [36]). Yoon et al. proposed Bayesian model-agnostic meta-learning (BMAML [25]). Lee et al. proposed Gradient-Based Meta-Learning with Learned Layerwise Metric and Subspace (MT-Net [7]).

2.2. Capsule Network

Inspired by the fact that manual features can effectively represent the position and posture information of an object, Hinton proposed the concept of the “capsule” [37] in 2011, which uses a vector to represent the capsule to increase the model’s ability to recognize gestures. After the pose or position of the same object changes, its existence probability remains unchanged, while the elements in the capsule change, that is, the covariance of object representation. Sabour [16] proposed a vector-form capsule network and a dynamic routing algorithm among capsules; using a 16-dimensional vector to represent the capsule, the modulo length of the vector represents the probability of an object’s existence, and the direction of the vector represents the attribute information of the object. In order to strengthen the ability of the capsule network to express covariance between objects, the “matrix capsule” [17] expands position sharing between different positions by adding a “local-global” relationship: the capsule is represented by a two-tuple composed of a pose matrix and an activation probability. The change in the angle of view usually has a complex relationship with the change in pixel intensity and has a simple linear relationship with the attitude matrix. The change in the angle of view can be better described by the change in the attitude matrix; that is, the covariance representation of the change in the angle of view can be obtained [38–40]. Objects are represented in the form of triples of the object’s existence probability [37], eigenvector, and pose matrix. Assuming that an object is composed of a series of components, an unsupervised capsule network autoencoder can be constructed to represent the individual components and the geometric relationship between them, thereby inferring the positional relationship between objects and objects.

The improvement of the vector capsule network mainly includes two aspects: the capsule form and routing design method. Aiming at the improvement of capsules, compared to the method of directly segmenting the extracted feature tensor with vector capsules, a method of orthogonal projection was proposed to obtain capsules [41–43]. For the im-

provement of routing, inspired by the better expressive ability of deep convolutional neural networks [44], Rajasegaran et al. [45] improved the original dynamic routing by using 3D convolution to construct a deeper network structure; DeCapsGAN [46] combined capsule and generative confrontation, in which the networks are combined with skip connections to increase the model depth. In order to strengthen the equivariant feature expression ability of the capsule network, Choi et al. [47] tried to improve the original dynamic routing by using attention routing and achieved good results in a perturbation interpretability experiment of capsule elements. The application aspects of capsule networks include tasks such as adversarial sample detection [48,49], point cloud segmentation [50], and 3D object recognition [51,52].

3. Mathematical Preliminaries

3.1. Model-Agnostic Meta-Learning

We start with an introduction to meta-learning in the context of few-shot learning. Assuming a task distribution $p(\tau)$, each task τ_i can be represented by a dataset D_i that consists of two disjoint sets: a support set D_i^S and a query set D_i^Q . Each set consists of some pairs of samples x and the labels y : $D_i^S = \{x_i^s, y_i^s\}_{s=1}^n$ and $D_i^Q = \{x_i^q, y_i^q\}_{q=1}^m$. In this work, we focus on gradient-based meta-learning [1], where we aim to learn a well-generalized parameter θ_0 of a base predictive learner f from N meta-learning tasks $\{\tau_i\}_{i=1}^N$. The meta-learning initialization for the base learner leads to bi-level optimization, with inner-loop optimization and outer-loop optimization. In inner-loop optimization, the base learner is initialized with θ_0 and adapted to the i -th task by updating its weight for a fixed number of steps via gradient descent with respect to the support set D_i^S .

$$\theta_i = \theta_0 - \alpha \nabla_{\theta} \mathcal{L}(D_i^S; \theta) \quad (1)$$

Here, α represents the inner-loop learning rate, and θ_0 denotes the initialized parameters of the base learner. During outer-loop optimization, the meta-learned initialization parameter θ_0 is evaluated by the generalization performance of the task-specific base learner with the parameter θ_i on the unseen query set D_i^Q . To evaluate and improve the initialization update, we measure the performance of θ_i on the unseen query set D_i^Q and use the corresponding loss to optimize the initialization as follows:

$$\theta_0^{k+1} = \theta_0^k - \beta \sum_{i=1}^N \mathcal{L}(D_i^Q; \theta_i) \quad (2)$$

where β is the outer-loop learning rate. In the outer-loop update, the meta-learner uses the gradients of all query sets in N tasks $\{\tau_i\}_{i=1}^N$ to update the parameters, that is, to find suitable initialization parameters for all query sets. After training for K time-step updates, we can obtain the well-generalization model parameter initialization θ_0^* . During the meta-test, θ_0^* is adapted to each task τ_i by performing a few gradient updates on the test task support set, and the final model performance is obtained on the corresponding query set.

3.2. Setting Up the System: Task-Covariant Representation Meta-Learning

In this section, we introduce the details of our proposed task-covariant representation meta-learning method. In order to better explain how our proposed model works, we use Figure 2, Algorithm 1, and Figure 3 to show the working principle of our model. The goal of task-covariant representation meta-learning is to effectively solve the multi-distribution problem of tasks by leveraging the transferable knowledge learned from historical tasks, with each distribution learning a modulated function. As shown in the task-covariant representation part of Figure 1, the adaptation is entirely determined by the real task distribution, which is represented by the learning of the capsule network. Task representation techniques are discussed, which aim to improve the efficiency of feature representation, the performance of task-invariant learning, and task-specific knowledge

adaptation. We also introduce the concept of task-covariant representation, which allows for adaptive task assignment to appropriate distributions when new tasks are introduced. Task-specific knowledge adaptive modulation involves choosing the suitable modulation technique based on the distribution of tasks while adjusting initialization parameters according to task-specific information.

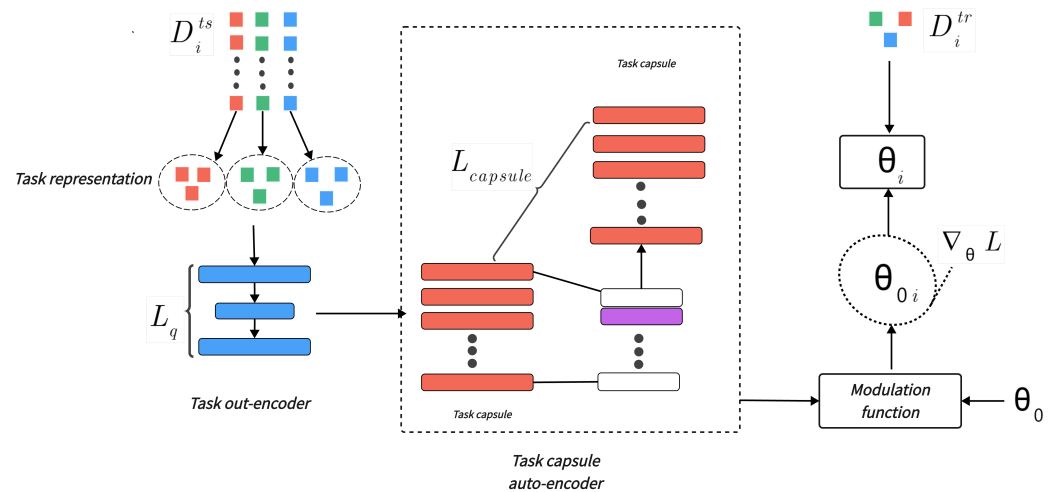


Figure 2. The framework for the task-covariant representation meta-learning algorithm.

Algorithm 1 Task-covariant representation for meta-learning.

- Input:** $p(\mathcal{T})$: Overall distribution of tasks; α : each task gradient update step size (inner-loop update); β : meta-optimization gradient update step size (outer-loop update); μ_1 and μ_2 : weight coefficients in the loss function.
- 1: Randomly initialize all learnable parameters Φ .
 - 2: **while** not done **do**
 - 3: Sample a batch of tasks $\{\mathcal{T}_i \mid i \in [1, I]\}$ from task distribution $p(\mathcal{T})$.
 - 4: **for** \mathcal{T}_i **do**
 - 5: Sample training set D_i^{tr} and testing set D_i^{ts} .
 - 6: Extract features to obtain task representation c_i^n by Equation (3).
 - 7: Extract deep task features based on recurrent neural network autoencoder T_i^n and compute class feature autoencoding loss L_q by Equation (4).
 - 8: Dynamic routing represents the covariant relationship between tasks, and the capsule decoder uses Equation (5) to calculate the task capsule reconstruction $L_{capsule}$.
 - 9: Compute the task-specific initialization θ_{0i} in Equation (7) and update parameters $\theta_0 = \theta_{0i} - \alpha \nabla L(f_{\theta}, D_i^{tr})$.
 - 10: **end for**
 - 11: Update parameters $\Phi \leftarrow \Phi - \beta \nabla_{\theta} \sum_{i=1}^I [L(f_{\theta}, D_i^{ts}) + \mu_1 L_q + \mu_2 L_{capsule}]$.
 - 12: **end while**

3.2.1. Task Representation Learning

To accurately reconcile the learned meta-knowledge, it is necessary to represent the current task information using covariance. When dealing with classification or regression problems in few-shot learning, we typically begin by investigating the task representation with respect to a set of samples. Initially, the samples in the task can be represented as vectors after feature extraction. To reduce the complexity of task representation in meta-learning, we typically average the feature vectors of samples belonging to the same class and then combine the resulting task feature vectors into a matrix format, where each row represents a class or cluster. In a regression problem, the image’s class information is replaced by clustering results. Constructing the task’s feature representation based on classes can effectively reduce the impact of outlier samples on tasks. Specifically, for a

classification problem, the support set $D_i^S = \{x_i^s, y_i^s\}_{s=1}^n$ of the corresponding task, with the class feature representation denoted by $c_i^n \in R$, is defined as:

$$c_i^n = \frac{1}{K_n^S} \sum_{j=1}^{K_n^S} F(x_j) \quad (3)$$

where K_n^S denotes the number of samples of the corresponding class in the support set of each task, and i represents the i -th task. $F(\cdot)$ is a feature extraction function that contains a convolutional layer and fully connected layer, which projects x_j into a feature space in which samples from the same class are located closer to each other, while samples from different classes are farther apart. For a regression problem, we first extract the feature of each sample and then cluster the extracted feature vectors to obtain the feature representation of the regression problem. The extracted task features are input into the recurrent neural network (RNN) autoencoder to extract the deep features of the task, where each class or cluster of features in the task is individually self-encoded to obtain a better task representation. In order to obtain better self-encoded intermediate features, we use the L_2 norm as the recurrent autoencoding loss function; the autoencoder loss function is defined as:

$$L_q = \sum_{n=1}^{N_k^S} \|c_i^n - c_i^{n*}\|_2 \quad (4)$$

where c_i^n and c_i^{n*} are the input and output of the autoencoder, and N_k^S is the number of support set classes or clusters.

From the above support-centralized sample feature extraction and class feature self-encoding operations, a better class feature representation T_i^n can be obtained, which prepares for subsequent task distribution division and the customization of specific task initialization parameters.

3.2.2. Task-Covariant Representation

In this section, we introduce task-covariant representation by leveraging the capsule network. Our goal is to use capsules for the covariant representation of tasks based on different task distributions. We take the intermediate variable generated by the encoder or the divided intermediate variable of the class feature vector in the task as a capsule; the class feature vector is denoted by T_i^n . The class feature vector is used as the input of the capsule network. First, projection transformation is performed to effectively represent covariant transformation among tasks. At the same time, the dimension of the class feature vector can be changed; the output of the capsule network consists of multiple capsules, and each capsule represents one of the multiple distributions. Capsules are assigned to positions that represent different distributions, and the value of the position where the capsule with the largest modulus length is retained, while other positions are masked (all set to 0) so that the task representation of the corresponding distribution can be obtained, as shown in Figure 4. In order to ensure that the task presentation information is not lost due to different mask operations, we transform the matrix composed of differently distributed capsules into a vector form and input it into the decoding network, which consists of three fully connected layers. The recurrent neural network self-encoder generates the intermediate variable T_i^n as the input of the capsule network. The output and decoding of the capsule network are $t_{capsule}$ and T_i^{n*} , respectively, and the L_2 norm is used as the loss function of the task capsule.

$$L_{capsule} = \|T_i^n - T_i^{n*}\|_2 \quad (5)$$

where T_i^n and T_i^{n*} are the capsule network input and decoding output. The capsule reconstruction error in Equation (5) can enhance the training stability, leading to the improvement of task representation learning.

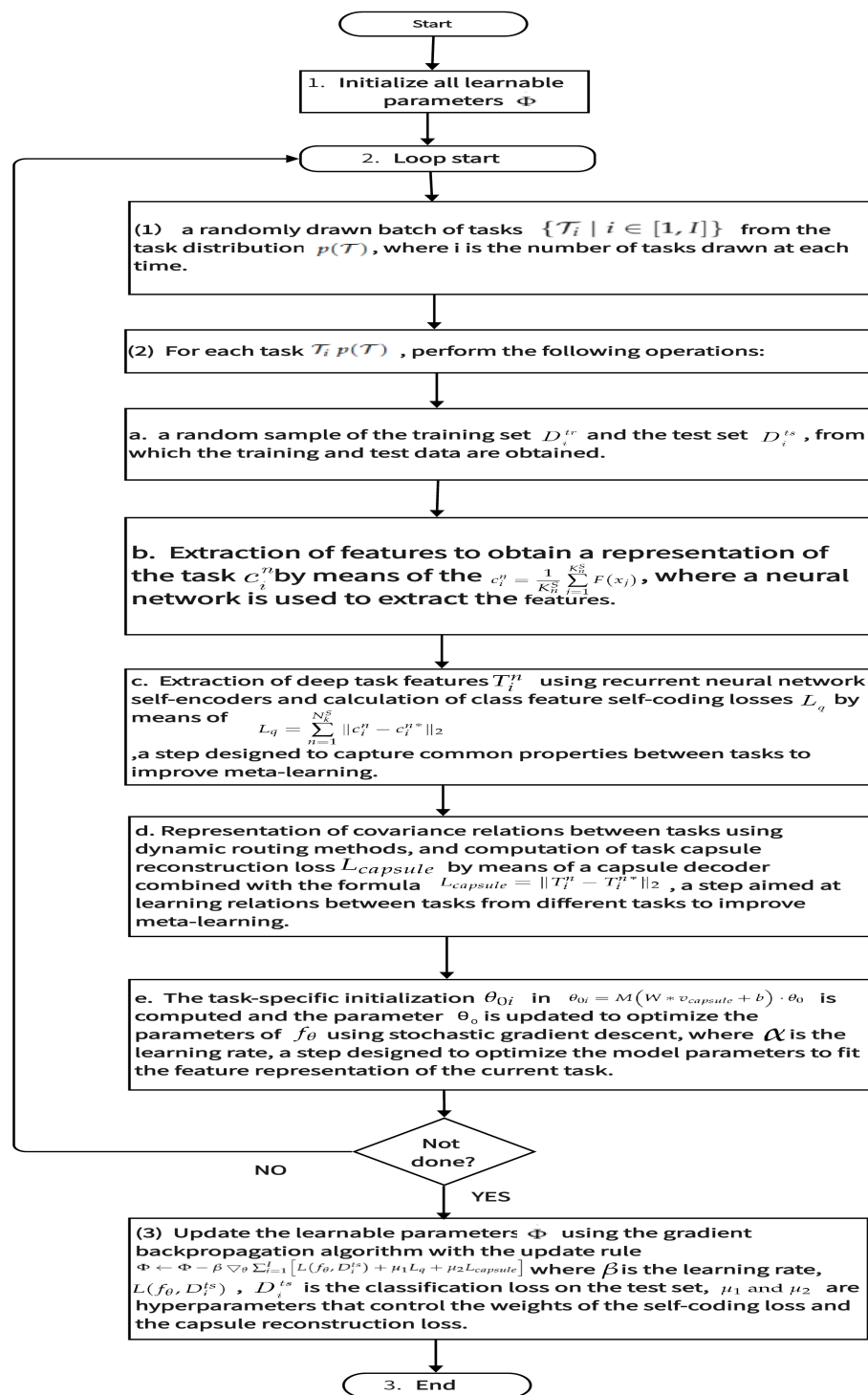


Figure 3. Algorithm flow diagram.

The output $t_{capsule}$ of the capsule network is transformed into a vector form $v_{capsule}$ after being masked by Equation (6), which helps the modulation function of the fully connected form to modulate the learned meta-knowledge.

$$v_{capsule} \leftarrow t_{capsule} \tag{6}$$

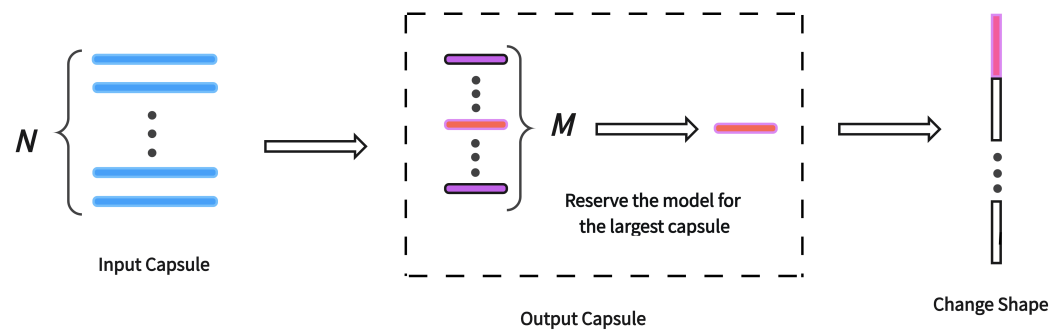


Figure 4. In the routing process of the capsule network, the number of input and output capsules are N and M , respectively; the mask mechanism is used so that the output capsule retains the largest capsule of the model, and the rest of the capsules are all zero.

3.2.3. Task-Specific Knowledge Adaption and Loss Function

We will discuss how to adjust the initialization parameters and loss function of a model corresponding to different distributions for each task. In the method developed in this study, we use modulation functions to tailor initialization parameters for particular tasks across diverse distributions. Differently distributed task capsules utilize distinct modulation functions.

The prerequisite for the effective operation of the modulation function is task-covariant representation. We use multi-distribution covariant representation to more effectively represent the true distribution of each task. The true distribution of tasks may take the form of multiple distributions. For each distribution, a modulation function is used. In other words, multiple tasks form a task set and share a modulation function. The number of task sets determines the number of modulation functions needed. To simplify the form of multiple modulation functions, a fully connected network and sigmoid activation function are used as the modulation function for initialization parameters. The input is a masked vector, where different mask positions are equivalent to using different modulation functions. Therefore, one network can perform multiple modulation functions. This is formulated as:

$$\theta_{0i} = M(W * v_{capsule} + b) \cdot \theta_0 \tag{7}$$

where W and b are a fully connect layer of weights and biases when using the sigmoid activation function. The trained model parameters are utilized as the starting point for the meta-learning model, with a number of tasks sampled from various relational datasets. The meta-learning model is trained to optimize the similarity vectors of tasks. When selecting a task, the current state of the meta-learning model is used as input, and a selection strategy is output based on the similarity vectors between the current task and the tasks previously encountered by the meta-learning model. This selection strategy is used to choose the next task, followed by retraining the meta-learning model to adapt to the new task.

For each task τ_i , we use the gradient descent algorithm to update the modulated initialization parameter θ_{0i} to θ_i . The total loss function includes meta-learning loss; class task reconstruction error, defined by Equation (4); and multi-distribution capsule reconstruction loss, defined by Equation (5). The total loss function is:

$$L_{total} = L_{meta} + \mu_1 L_q + \mu_2 L_{capsule} \tag{8}$$

where μ_1 and μ_2 are the proportions of task representation autoencoder loss and multi-distribution capsule reconstruction loss in the total loss. The goal of our neural network training is to minimize the total loss. Finally, we also provide a detailed table of common mathematical symbols, which can be referred to in Appendix A Table A1 if necessary.

4. Experiments

In this section, we describe experiments performed on several few-shot learning problems, such as few-shot regression, few-shot classification, and ablation experiments to corroborate the effectiveness of task-covariant representation meta-learning. All our experiments were supervised small-sample learning problems that did not use pre-trained models.

4.1. Two-Dimensional Regression

To demonstrate the effectiveness of our proposed task-covariant meta-learning algorithm, we evaluated our method and other methods in a few-shot regression experiment. To verify the superiority of the model for more complex regression problems, AID, UCMerced-LandUse, and WHU-RS-19 were combined into a single dataset, and then 2D regression tests were performed on these data. Inputs x and y were sampled from a uniform distribution $U [0.0, 0.5]$, and random Gaussian noise with a standard deviation of 0.3 was added to the output. The regression functions include line, sinusoidal, quadratic, cubic quadratic surface, and ripple functions. Compared with task-invariant meta-learning algorithms, our proposed task-covariant meta-learning method has notable advantages in regression tasks and can still achieve competitive results compared with task-heterogeneous meta-learning methods. The experimental results are shown in Table 1.

Table 1. Comparison of experimental error (mean square error with 95% confidence) results between the task-covariant meta-learning algorithm and other meta-learning algorithms in 2D regression tasks.

Model	MAML	Meta-SGD	BMAML	MT-Net	Our	Enhancement
10-shot	0.89	1.05	0.65	0.68	0.45	0.26
7-shot	0.93	1.08	0.67	0.70	0.48	0.24
5-shot	1.0	1.12	0.70	0.75	0.56	0.14

4.2. Few-Shot Classification

In a few-shot classification problem, each task is defined by N -way K -shot classification, where N is the number of classes, and K is the number of examples per class.

4.2.1. Datasets and Setting

In the few-shot classification task, we used common remote sensing image datasets, which mainly include AID, SIRI-WHU, UCMerced-LandUse, and WHU-RS-19. In these experiments, the size of the data images was made uniform at 84×84 , and each dataset was divided into a support set and query set at a ratio of 8:2, with each experiment being run for 4000 epochs. First, we conducted experiments on the classification of AID and UCMerced-LandUse datasets. We compared our proposed method not only internally but also with other methods. To demonstrate the effectiveness of our proposed task-covariant meta-learning in handling more complex underlying structures and highlight its application ability for remote sensing images, we increased the difficulty of few-shot image classification by adding synthetic data that introduce blur and sharpening to the original AID dataset. The processed datasets were used separately to conduct experiments in the few-shot classification task. Finally, to validate the performance of our method on other datasets, experiments were carried out on UCMerced-LandUse and WHU-RS-19 datasets, and the results were compared with task-invariant meta-learning (MAML) and task-heterogeneous meta-learning (MeTAL) methods.

4.2.2. Internal Comparison of Our Method

We verified the effectiveness of our proposed scheme through a large number of experiments, the results of which are shown in Table 2. The experiments tested the following settings: (a) The type of image corresponding to the task is directly input to the encoder to generate intermediate features, which are then directly input to the modulation function. This exploits the heterogeneity between tasks to customize initialization parameters.

(b) Building on experiment (a), the self-encoder result is first used with a fully connected layer, and then the initialization parameters are modulated. (c) The task-specific features are directly input to the capsule network, which has no reconstruction loss and produces a single output capsule. The initialization parameters are customized for each task using the output of the capsule network and modulation functions. (d) Similar to (c), but the number of output capsules is 5, and all other settings remain unchanged. (e) Similar to (c), but the number of output capsules is 10, and all other settings remain unchanged. (f) Similar to (c), but the number of output capsules is 15, and all other settings remain unchanged. (g) A fully connected neural network layer and capsule routing are used simultaneously, and then the modulation function is applied to customize the initialization parameters for a specific task. (h) Similar to (g), but a three-layer fully connected network is used. (i) Similar to (g), but each type of class task feature is separately encoded using the fully connected network.

Table 2. Comparison of classification accuracy (accuracy \pm 95%) between different settings of task-covariant meta-learning algorithm and other similar works on the AID and UCMerced-LandUse datasets.

AID			
Algorithm	5-way 1-shot	5-way 3-shot	5-way 5-shot
(a) Initialization parameters	78.47%	81.91%	85.58%
(b) Adjusting parameters with FCN	76.31%	80.96%	83.50%
(c) Number of capsules is 1	79.43%	82.19%	85.55%
(d) Number of capsules is 5	78.33%	81.80%	85.07%
(e) Number of capsules is 10	78.23%	82.10%	84.42%
(f) Number of capsules is 15	77.35%	80.93%	84.93%
(g) Adding a layer of FCN	77.69%	80.85%	82.45%
(h) Adding three layers of FCN	78.45%	81.65%	83.23%
(i) Coding each task separately	78.65%	82.14%	85.90%
UCMerced-LandUse			
(a) Initialization parameters	75.23%	82.61%	84.12%
(b) Adjusting parameters with FCN	73.59%	79.59%	83.03%
(c) Number of capsules is 1	75.29%	82.74%	84.53%
(d) Number of capsules is 5	74.73%	82.06%	83.91%
(e) Number of capsules is 10	73.59%	81.47%	84.36%
(f) Number of capsules is 15	75.07%	80.88%	82.84%
(g) Adding a layer of FCN	74.89%	81.01%	83.16%
(h) Adding three layers of FCN	75.29%	81.83%	83.74%
(i) Coding each task separately	75.37%	82.37%	84.03%

Experiment analysis: (1) From experimental settings (a) and (b), it can be seen that directly using the fully connected neural network to extract features as the class feature mean and then using the modulation function reduces the model's performance. (2) From experiments (c–f), it can be observed that as the number of capsules (i.e., the number of distributions of multi-distribution tasks) increases, the model's performance decreases. However, there is almost no difference between using 10 and 20 capsules. (3) From experiment (f) and experiments (g–i), it can be observed that extracting features from the intermediate coding of class features will reduce the performance of the model. (4) It can be seen from experiments (g) and (h) that the number of layers of the fully connected network has almost no effect on the experimental results, and (g) and (i) reveal that the fully connected neural network's separate coding can slightly improve the performance of the model. Through the above experimental analysis, we determined the final model as the form from experiment (a). The following sections introduce the experimental comparison between our proposed algorithm and other similar algorithms.

4.2.3. Comparison with Other Methods

Based on the experimental results obtained with different settings, we always selected setting (a) as the final model and compared its performance with other similar methods. We compared our proposed task-covariant meta-learning method with different types of baselines: (1) the task-invariant representation method in the original gradient-based learning method (all tasks share global initialization parameters, such as MAML and Meta-SGD) and the task-heterogeneous representation method (adaptive custom initialization parameters for different tasks, such as MT-Net, MUMO-MAML, HSML, and DMAML); (2) other original learning algorithms, including task-invariant representation methods (VERSA, Prototypical network, and TapNet) and task-heterogeneous representation (TADAM) as baselines.

The experimental results are shown in Table 3, in which our proposed task-covariant learning method is compared with other schemes. From the experimental data, it is clear that in the few-shot classification task, our method has a slight advantage over the other solutions. At the same time, we can find that our method has some advantages over global parameter-sharing meta-learning for task-covariant learning for the classification of real data with fewer samples. The results show that adaptively customizing the initialization parameters for each task according to the task information is more in line with the actual situation than the global sharing setting. In contrast to independently customizing each parameter, task covariance means that both the invariance between tasks and the heterogeneity between tasks are considered. However, the large amount of calculation required for the meta-knowledge graph is computationally intensive; the method of using the capsule network can significantly reduce the amount of calculation and has a higher training efficiency, and the running time is more stable. To verify the operational efficiency of our proposed method, we compared the ARML method with our method on the AID dataset. For task-covariant learning and the ARML method, under the condition that other settings remain the same, the running times of each iteration were compared (the average was taken once for 100 iterations, and all results were averaged again), and the results are shown in Table 4. Compared to RAML, the advantage of the proposed method in operating efficiency is significant, resulting in substantial time savings during training.

In order to compare the accuracy of trials during the training of task covariance learning and the ARML method, the following experimental settings were used: AID dataset and 5-way 5-shot. The verification accuracy of the two is drawn in the graph as scatter points, where the red dots are task-covariant learning, and the cyan dots are task-heterogeneous meta-learning. As can be seen in Figure 5, after a sufficient training process, the validation accuracy of both methods is similar, but our method also has a smaller range of fluctuations.

At the same time, in order to verify the effectiveness of our method, we added blur to and sharpened the AID dataset images to produce a degraded dataset. In the setup of the fuzzy dataset, a Gaussian fuzzy method was used with a radius value of 3, and the original default bound value was used. When sharpening the dataset, we used horizontal sharpening with the depth value set to -1 and the convolutional kernel matrix set to B .

$$B = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & -1 \end{bmatrix} \quad (9)$$

On the degraded AID dataset, the above methods and our method were compared on a small sample classification task; the experiments are shown in Table 5. We found that, on the degraded dataset, although the performance of our method decreased, it still exhibited some advantages over the other methods. This demonstrates that the capsule network's ability to provide a covariant representation of tasks can effectively enhance the expressive power of the tasks.

Table 3. Comparison of few-shot classification accuracy of task-covariant learning, task-invariant, and task-heterogeneous methods on AID and UCMerced-LandUse datasets.

AID			
Algorithm	5-way 1-shot	5-way 3-shot	5-way 5-shot
VERSA	68.58%	72.40%	75.86%
ProtoNet	70.11%	73.28%	77.67%
TapNet	70.90%	73.91%	79.07%
TADAM	69.58%	75.60%	79.13%
MAML	66.94%	72.01%	78.52%
Meta-SGD	68.58%	74.95%	77.87%
BMAML	67.89%	73.39%	79.01%
MT-Net	71.72%	77.54%	79.22%
MUMOMAML	69.82%	75.73%	80.49%
HSML	73.98%	79.84%	81.68%
Proposed	79.27%	81.91%	85.90%
Enhancement	5.29%	2.07%	4.22%

UCMerced-LandUse			
Algorithm	5-way 1-shot	5-way 3-shot	5-way 5-shot
VERSA	67.43%	72.81%	73.46%
ProtoNet	68.52%	74.62%	80.21%
TapNet	69.44%	74.56%	80.54%
TADAM	68.34%	74.70%	79.78%
MAML	68.66%	73.61%	78.56%
Meta-SGD	68.38%	74.31%	81.49%
BMAML	69.53%	75.50%	80.06%
MT-Net	68.80%	74.27%	82.57%
MUMOMAML	70.81%	75.36%	81.89%
HSML	71.01%	77.91%	82.08%
Proposed	75.23%	82.61%	84.12%
Enhancement	4.22%	4.70%	2.06%

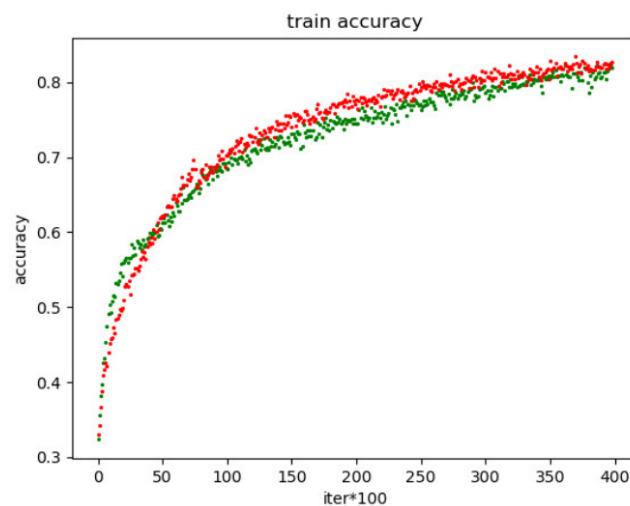


Figure 5. Comparison of accuracy between our proposed method and a task-heterogeneous meta-learning process, where the red dots represent task-covariant learning and the cyan dots represent task-heterogeneous meta-learning.

Table 4. Run time comparison.

Dataset	Algorithm	Time (Minutes)		
		5-Way 1-Shot	5-Way 3-Shot	5-Way 5-Shot
AID	Proposed	2.85	3.73	4.39
	ARML	3.06	4.07	5.42
	Enhancement	0.21	0.34	1.02
UCMerced-LandUse	Proposed	3.01	4.12	4.82
	ARML	4.10	4.92	5.47
	Enhancement	1.09	0.8	0.65

Table 5. Comparison of the few-shot classification accuracy of task-covariant learning, task-invariant, and task-heterogeneous methods on the degraded AID dataset.

Setting	Algorithm	Avg. Original	Avg. Blur	Avg. Sharpened
5-way 1-shot	VERSA	68.58%	65.98%	60.70%
	ProtoNet	70.11%	64.51%	58.24%
	TapNet	70.90%	65.16%	59.25%
	TADAM	69.58%	66.44%	61.02%
	MAML	66.94%	64.53%	58.71%
	Meta-SGD	69.58%	66.36%	62.21%
	BMAML	67.89%	65.08%	60.70%
	MT-Net	71.72%	64.64%	59.05%
	MUMOMAML	69.82%	66.59%	61.24%
	HSML	73.89%	64.62%	61.78%
	Proposed	79.27%	72.07%	66.55%
	Enhancement	5.29%	7.45%	4.77%
5-way 3-shot	VERSA	72.40%	70.10%	70.48%
	ProtoNet	73.28%	69.25%	68.34%
	TapNet	73.91%	70.24%	69.03%
	TADAM	75.60%	72.46%	71.78%
	MAML	72.01%	70.83%	68.04%
	Meta-SGD	74.95%	71.36%	70.37%
	BMAML	73.39%	69.84%	69.57%
	MT-Net	77.54%	73.69%	70.62%
	MUMOMAML	75.73%	70.23%	71.21%
	HSML	79.84%	72.17%	73.16%
	Proposed	81.91%	78.52%	77.49%
	Enhancement	2.07%	6.35%	4.33%
5-way 5-shot	VERSA	75.86%	75.41%	71.93%
	ProtoNet	77.67%	75.07%	72.15%
	TapNet	79.07%	75.21%	71.68%
	TADAM	79.13%	77.36%	75.15%
	MAML	78.52%	74.93%	71.59%
	Meta-SGD	77.82%	75.54%	72.24%
	BMAML	79.01%	76.21%	73.22%
	MT-Net	79.22%	76.65%	71.18%
	MUMOMAML	80.49%	78.29%	73.9%
	HSML	81.68%	78.93%	77.27%
	Proposed	85.90%	80.14%	80.42%
	Enhancement	4.22%	1.21%	3.15%

4.2.4. SIRI-WHU and WHU-RS19 Datasets

To confirm that the task-covariant representation meta-learning method is also advantageous using other widely used remote sensing image classification datasets, two commonly employed datasets for few-shot classification, SIRI-WHU and WHU-RS19, were utilized. The experimental results presented in Table 6 demonstrate that, compared with task-invariant meta-learning (MAML) and task-heterogeneous meta-learning (MeTAL, which involves the adaptive learning of the loss function for each task using task-specific information) methods, our method still exhibited superior performance. However, our classification accuracy was slightly lower than that of the MeTAL method.

Table 6. Comparison of classification accuracy on SIRI-WHU and WHU-RS19 datasets.

SIRI-WHU			
Model	1-shot	3-shot	5-shot
MAML	68.52%	75.84%	79.06%
MAML++	72.12%	81.59%	83.15%
MeTAL	77.48%	85.40%	86.40%
proposed	78.83%	83.57%	85.02%
Enhancement	1.3%	−1.83%	−1.83%
WHU-RS19			
MAML	74.63%	83.79%	87.75%
MAML++	78.57%	86.23%	88.95%
MeTAL	81.96%	89.93%	92.41%
proposed	84.63%	90.05%	91.75%
Enhancement	2.2%	0.21%	−0.66%

5. Conclusions

In this paper, we propose a task-covariant meta-learning method for representation learning specifically designed for remotely sensed images. The proposed approach tackles the challenge of assigning the current task to a specific subspace and acquiring the corresponding modulation function for each respective subspace. The task-covariant representation meta-learning algorithm incorporates capsule representation to embody a subset of the task feature or the entire task feature. Additionally, the dynamic routing algorithm is leveraged to capture the covariant relationships among tasks. By learning distinct modulation functions for each distribution within a multi-distribution task, the modulation functions are able to be dynamically selected based on the specific distributions encountered during the task. Logically, our method can effectively handle the real task distribution, and numerous experiments showed that the proposed new meta-learning algorithm can achieve accurate results comparable to those of state-of-the-art algorithms and effectively improve model training performance. It is expected that the ideas and structure of the task-covariant representation meta-learning method would make a valuable contribution to future research in meta-learning. To fulfill the research demands in the domain of future image enhancement, it is crucial to bridge technological and disciplinary boundaries and integrate knowledge from various interdisciplinary fields, including machine learning, computer vision, and sensor technology, in order to consistently advance the growth of image enhancement algorithms and their corresponding applications. Therefore, in future work, we aim to explore the use of capsule networks and meta-learning in a weakly [53] supervised manner to achieve image enhancement based on human visual perception.

Author Contributions: Conceptualization, L.Z. and Z.T.; methodology, L.Z.; software, L.Z.; validation, L.Z., Y.T., Z.J. and Z.T.; formal analysis, Y.T.; investigation, L.Z.; resources, Z.T.; data curation, Z.J.; writing—original draft preparation, Y.T.; writing—review and editing, L.Z.; visualization, L.Z.; supervision, L.Z.; project administration, Y.T. and Z.J.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 61866040), the Special Support Plan for High-Level Talents of Guangdong Province (No. 2019TQ05X571), and the Project of Guangdong Province Innovative Team (No. 2020WCXTD011).

Data Availability Statement: The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of 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.

Appendix A. Commonly Used Notations

Table A1. Commonly used notations.

Notation	Description
D_i^S	The support set of the i -th task, containing samples and labels used for training the model.
D_i^Q	The query set of the i -th task, containing samples and labels for testing the model.
x_i^s	A sample from the support set of the i -th task.
y_i^s	The label of a sample from the support set of the i -th task.
x_i^q	A sample from the query set of the i -th task.
y_i^q	The label of a sample from the query set of the i -th task.
θ_0	The initial parameter of the base predictive model f , which is the goal of meta-learning.
$\{\tau_i\}_{i=1}^N$	A set of N meta-learning tasks used for training the model.
θ_i	The model parameter for the i -th task.
α	The learning rate of the inner optimization strategy.
$\nabla_{\theta} \mathcal{L}(D_i^S; \theta)$	The gradient with respect to θ , representing the gradient of the model trained on the support set D_i^S with respect to the current parameter.
θ_i	Parameters of the task-specific base learner.
\mathcal{L}	Loss function.
β	Step size or learning rate for outer optimization loop.
R	A set of real numbers.
$\frac{1}{K_n^S}$	Fraction equal to the reciprocal of the number of samples in a support set.
Σ	The summation of all terms within the brackets.
j	The index used for iterating over all samples in a support set.
$F(\cdot)$	The feature extraction function.
x_j	The input sample.
i	The index used for iterating over all tasks.
c_i^n	The class feature representation for i -th task.
K_n^S	The number of samples of the corresponding class in the support set of each task.

Table A1. Cont.

Notation	Description
L_2 norm	Euclidean distance between two points.
μ_1 and μ_2	The weight coefficients that control the ratio of autoencoder loss and capsule network reconstruction loss in the loss function.
Φ	The set of learnable parameters in the model, including the weights and biases of the neural network.
$\{\mathcal{T}_i \mid i \in [1, I]\}$	The set of I tasks sampled from the meta-task distribution.
D_i^{tr} and D_i^{ts}	The training set and testing set of the i -th task.
c_i^n	A feature vector representing task i .
T_i^n	An autoencoder recurrent neural network used to extract task-specific low-dimensional feature vectors for task i .
L_q	The loss function used to ensure consistency in task-specific initialization.
$L_{capsule}$	The loss function for computing task-specific capsule network reconstruction.
θ_{0_i}	The function parameters used for task-specific initialization in task i .
$\nabla L(f_\theta, D_i^{tr})$	The task-specific gradient computation used to update task-specific initialization.
∇_θ	The meta-learning update gradient computation used to update the learnable parameters of the model.
$\ \cdot\ _2$	Represents the Euclidean norm or 2-norm.
N_k^S	Indicates the number of support set categories or clusters.
$t_{capsule}$	The output of the capsule network.
T_i^{n*}	The decoding output of the capsule network.
$\ \cdot\ _2$	The Euclidean norm or 2-norm.
$v_{capsule}$	The vector form of the output of capsule network.
M	The modulation function.
W and b	The weights and biases of a fully connected layer.
L_{meta}	The meta-learning loss.

References

1. Finn, C.; Abbeel, P.; Levine, S. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Precup, D., Teh, Y.W., Eds.; Cornell University Library: New York, NY, USA, 2017; Volume 70, pp. 1126–1135.
2. Finn, C.; Xu, K.; Levine, S. Probabilistic Model-Agnostic Meta-Learning. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2018; Volume 31.
3. Gwilliam, M.; Shrivastava, A. Beyond Supervised vs. Unsupervised: Representative Benchmarking and Analysis of Image Representation Learning. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 9632–9642. [[CrossRef](#)]
4. Vuorio, R.; Sun, S.H.; Hu, H.; Lim, J.J. Toward Multimodal Model-Agnostic Meta-Learning. In Proceedings of the 2nd Workshop on Meta-Learning at the Thirty-Second Annual Conference Neural Information Processing Systems, Vancouver, BC, Canada, 8 December 2019.
5. Yao, H.; Wei, Y.; Huang, J.; Li, Z. Hierarchically Structured Meta-learning. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; Volume 97, pp. 7045–7054.

6. Jha, A.; Bose, S.; Banerjee, B. GAF-Net: Improving the Performance of Remote Sensing Image Fusion Using Novel Global Self and Cross Attention Learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–7 January 2023; pp. 6354–6363.
7. Lee, Y.; Choi, S. Gradient-Based Meta-Learning with Learned Layerwise Metric and Subspace. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Dy, J., Krause, A., Eds.; Volume 80, pp. 2927–2936.
8. Oreshkin, B.; Rodríguez López, P.; Lacoste, A. TADAM: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2018; Volume 31.
9. Yao, H.; Wu, X.; Tao, Z.; Li, Y.; Ding, B.; Li, R.; Li, Z. Automated Relational Meta-learning. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
10. Ji, J.Y.; Wong, M.L. Surrogate-assisted Parameter Re-initialization for Differential Evolution. In Proceedings of the 2022 IEEE Symposium Series on Computational Intelligence (SSCI), Singapore, 4–7 December 2022; pp. 1592–1599. [[CrossRef](#)]
11. Baik, S.; Choi, J.; Kim, H.; Cho, D.; Min, J.; Lee, K.M. Meta-Learning With Task-Adaptive Loss Function for Few-Shot Learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 9465–9474.
12. Antoniou, A.; Edwards, H.; Storkey, A. How to train your MAML. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
13. Baik, S.; Choi, M.; Choi, J.; Kim, H.; Lee, K.M. Meta-Learning with Adaptive Hyperparameters. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: New York, NY, USA, 2020; Volume 33, pp. 20755–20765.
14. Huaxiu, Y.; Yu, W.; Ying, W.; Peilin, Z.; Mehrdad, M.; Defu, L.; Chelsea, F. Meta-learning with an Adaptive Task Scheduler. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2021; Volume 35.
15. Kaddour, J.; Saemundsson, S.; Deisenroth (he/him), M. Probabilistic Active Meta-Learning. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: New York, NY, USA, 2020; Volume 33, pp. 20813–20822.
16. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
17. Hinton, G.E.; Sabour, S.; Frosst, N. Matrix capsules with EM routing. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
18. Kosíorek, A.; Sabour, S.; Teh, Y.W.; Hinton, G.E. Stacked Capsule Autoencoders. In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2019; Volume 32.
19. Yu, C.; Zhu, X.; Zhang, X.; Wang, Z.; Zhang, Z.; Lei, Z. HP-Capsule: Unsupervised Face Part Discovery by Hierarchical Parsing Capsule Network. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 4022–4031. [[CrossRef](#)]
20. Thrun, S.; Pratt, L. Learning to Learn: Introduction and Overview. In *Learning to Learn*; Thrun, S., Pratt, L., Eds.; Springer: Boston, MA, USA, 1998; pp. 3–17. [[CrossRef](#)]
21. Wang, L.; Zhang, S.; Han, Z.; Feng, Y.; Wei, J.; Mei, S. Diversity Measurement-Based Meta-Learning for Few-Shot Object Detection of Remote Sensing Images. In Proceedings of the IGARSS 2022—2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022; pp. 3087–3090. [[CrossRef](#)]
22. Ravi, S.; Larochelle, H. Optimization as a Model for Few-Shot Learning. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
23. Metz, L.; Maheswaranathan, N.; Cheung, B.; Sohl-Dickstein, J. Learning Unsupervised Learning Rules. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
24. Liang, B.; Wang, X.; Wang, L. Impact of Heterogeneity on Network Embedding. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1296–1307. [[CrossRef](#)]
25. Yoon, J.; Kim, T.; Dia, O.; Kim, S.; Bengio, Y.; Ahn, S. Bayesian Model-Agnostic Meta-Learning. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2018; Volume 31.
26. Grant, E.; Finn, C.; Levine, S.; Darrell, T.; Griffiths, T. Recasting Gradient-Based Meta-Learning as Hierarchical Bayes. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
27. Baik, S.; Hong, S.; Lee, K.M. Learning to Forget for Meta-Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
28. Bengio, Y.; Louradour, J.; Collobert, R.; Weston, J. Curriculum Learning. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; Association for Computing Machinery: New York, NY, USA, 2009; pp. 41–48. [[CrossRef](#)]
29. Wang, X.; Chen, Y.; Zhu, W. A Survey on Curriculum Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [[CrossRef](#)] [[PubMed](#)]

30. Canevet, O.; Fleuret, F. Large Scale Hard Sample Mining With Monte Carlo Tree Search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
31. Shrivastava, A.; Gupta, A.; Girshick, R. Training Region-Based Object Detectors With Online Hard Example Mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
32. Sun, Q.; Liu, Y.; Chen, Z.; Chua, T.S.; Schiele, B. Meta-Transfer Learning through Hard Tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [[CrossRef](#)] [[PubMed](#)]
33. Sun, Q.; Liu, Y.; Chua, T.S.; Schiele, B. Meta-Transfer Learning for Few-Shot Learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–17 June 2019; pp. 403–412. [[CrossRef](#)]
34. Liu, C.; Wang, Z.; Sahoo, D.; Fang, Y.; Zhang, K.; Hoi, S.C.H. Adaptive Task Sampling for Meta-learning. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 752–769.
35. Zheng, X.; Chen, X.; Lu, X. Visible-Infrared Person Re-Identification via Partially Interactive Collaboration. *IEEE Trans. Image Process.* **2022**, *31*, 6951–6963. [[CrossRef](#)] [[PubMed](#)]
36. Li, Z.; Zhou, F.; Chen, F.; Li, H. Meta-SGD: Learning to Learn Quickly for Few Shot Learning. *arXiv* **2017**, arXiv:1707.09835.
37. Hinton, G.E.; Krizhevsky, A.; Wang, S.D. Transforming Auto-Encoders. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2011, Espoo, Finland, 14–17 June 2011; Honkela, T., Duch, W., Girolami, M., Kaski, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 44–51.
38. Cohen, T.S.; Welling, M. Group Equivariant Convolutional Networks. In Proceedings of the 33rd International Conference on International Conference on Machine Learning—Volume 48, New York, NY, USA, 19–24 June 2016; pp. 2990–2999.
39. Dieleman, S.; Fauw, J.D.; Kavukcuoglu, K. Exploiting Cyclic Symmetry in Convolutional Neural Networks. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Balcan, M.F., Weinberger, K.Q., Eds.; Volume 48, pp. 1889–1898.
40. Worrall, D.E.; Garbin, S.J.; Turmukhambetov, D.; Brostow, G.J. Harmonic Networks: Deep Translation and Rotation Equivariance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
41. Zhang, L.; Edraki, M.; Qi, G.J. CapProNet: Deep Feature Learning via Orthogonal Projections onto Capsule Subspaces. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2018; Volume 31.
42. Edraki, M.; Rahnnavard, N.; Shah, M. SubSpace Capsule Network. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10745–10753.
43. Wang, X.; Wang, Y.; Guo, S.; Kong, L.; Cui, G. Capsule Network With Multiscale Feature Fusion for Hidden Human Activity Classification. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–12. [[CrossRef](#)]
44. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
45. Rajasegaran, J.; Jayasundara, V.; Jayasekara, S.; Jayasekara, H.; Seneviratne, S.; Rodrigo, R. DeepCaps: Going Deeper With Capsule Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–17 June 2019.
46. Lyu, Q.; Guo, M.; Ma, M.; Mankin, R. DeCapsGAN: Generative adversarial capsule network for image denoising. *J. Electron. Imaging* **2021**, *30*, 033016. [[CrossRef](#)]
47. Choi, J.; Seo, H.; Im, S.; Kang, M. Attention Routing Between Capsules. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 1981–1989. [[CrossRef](#)]
48. Qin, Y.; Frosst, N.; Sabour, S.; Raffel, C.; Cottrell, G.; Hinton, G. Detecting and Diagnosing Adversarial Images with Class-Conditional Capsule Reconstructions. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
49. Gu, J.; Wu, B.; Tresp, V. Effective and Efficient Vote Attack on Capsule Networks. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 3–7 May 2021.
50. Sun, W.; Tagliasacchi, A.; Deng, B.; Sabour, S.; Yazdani, S.; Hinton, G.; Yi, K.M. Canonical Capsules: Self-Supervised Capsules in Canonical Pose. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: New York, NY, USA, 2021; Volume 34.
51. Sun, K.; Zhang, J.; Liu, J.; Yu, R.; Song, Z. DRCNN: Dynamic Routing Convolutional Neural Network for Multi-View 3D Object Recognition. *IEEE Trans. Image Process.* **2021**, *30*, 868–877. [[CrossRef](#)] [[PubMed](#)]
52. Jenkins, P.; Armstrong, K.; Nelson, S.; Gotad, S.; Jenkins, J.S.; Wilkey, W.; Watts, T. CountNet3D: A 3D Computer Vision Approach to Infer Counts of Occluded Objects. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–8 January 2023; pp. 3007–3016. [[CrossRef](#)]
53. Zheng, X.; Zhang, Y.; Zheng, Y.; Luo, F.; Lu, X. Abnormal event detection by a weakly supervised temporal attention network. *CAAI Trans. Intell. Technol* **2022**, *7*, 419–431. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.