



Multimodal meta-learning through meta-learned task representations

Anna Vettoruzzo¹ · Mohamed-Rafik Bouguelia¹ · Thorsteinn Rögnvaldsson¹

Received: 27 July 2023 / Accepted: 22 January 2024 / Published online: 23 February 2024
© The Author(s) 2024

Abstract

Few-shot meta-learning involves training a model on multiple tasks to enable it to efficiently adapt to new, previously unseen tasks with only a limited number of samples. However, current meta-learning methods assume that all tasks are closely related and belong to a common domain, whereas in practice, tasks can be highly diverse and originate from multiple domains, resulting in a multimodal task distribution. This poses a challenge for existing methods as they struggle to learn a shared representation that can be easily adapted to all tasks within the distribution. To address this challenge, we propose a meta-learning framework that can handle multimodal task distributions by conditioning the model on the current task, resulting in a faster adaptation. Our proposed method learns to encode each task and generate task embeddings that modulate the model's activations. The resulting modulated model become specialized for the current task and leads to more effective adaptation. Our framework is designed to work in a realistic setting where the mode from which a task is sampled is unknown. Nonetheless, we also explore the possibility of incorporating auxiliary information, such as the task-mode-label, to further enhance the performance of our method if such information is available. We evaluate our proposed framework on various few-shot regression and image classification tasks, demonstrating its superiority over other state-of-the-art meta-learning methods. The results highlight the benefits of learning to embed task-specific information in the model to guide the adaptation when tasks are sampled from a multimodal distribution.

Keywords Meta-learning · Few-shot learning · Transfer learning · Task representation · Multimodal distribution

1 Introduction

Humans possess a remarkable ability to learn new tasks using only a few examples by leveraging their prior experience and context, and quickly adapting to novel situations. In contrast, conventional deep learning methods are designed for specific tasks, which limits their performance in terms of data efficiency and generalization. Like humans, it is desirable for learning algorithms to be able to adapt efficiently to new tasks and incorporate new

information to improve their performance. Meta-learning achieves this by learning a representation or acquiring general knowledge from multiple tasks during meta-training and adapting it to new tasks at meta-testing time. Specifically, optimization-based meta-learning methods aim to learn an initialization of neural network parameters that can be efficiently fine-tuned for new tasks with only a few training examples and gradient updates. Current methods typically assume that all tasks are related and rely on a shared model initialization for all tasks. While this approach may work well in some cases, it may not be appropriate for complex scenarios where tasks are heterogeneous and drawn from a multimodal distribution with multiple unknown modes. In such scenarios, fine-tuning the same initial model to any task using only a few examples may lead to suboptimal results. To illustrate, when learning a Latin language like Spanish, humans can leverage not only their prior experience of reading, writing or listening in that language (i.e., tasks from the same mode) but also their knowledge of related languages such as Italian and French (i.e., tasks from other similar modes). However, exploiting knowledge from entirely different languages

Anna Vettoruzzo and Mohamed-Rafik Bouguelia have contributed equally to this work.

✉ Anna Vettoruzzo
anna.vettoruzzo@hh.se

Mohamed-Rafik Bouguelia
mohamed-rafik.bouguelia@hh.se

Thorsteinn Rögnvaldsson
thorsteinn.rognvaldsson@hh.se

¹ Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Halmstad, Sweden

like Chinese or Arabic (i.e., modes less similar to the previous one) may be less helpful. Additionally, tasks unrelated to language learning, such as drawing or painting (i.e., from a mode completely disjoint and far apart from the previous ones), may not provide any useful knowledge for learning Spanish.

Model Agnostic Meta-Learning (MAML) [1] is a well-known and widely used method in meta-learning that has demonstrated impressive performance in few-shot learning scenarios. The fundamental objective of MAML is to learn an optimal set of initial model parameters that can be effectively fine-tuned on new tasks with a small number of examples. This is accomplished using a set of training tasks, allowing the initial model to achieve good generalization performance on any task after only a few gradient descent steps on that task. This method is effective when all tasks belong to the same domain (i.e., drawn from a unimodal task distribution). However, its performance suffers when tasks are heterogeneous and drawn from multiple different domains or modes (e.g., tasks related to classifying digits from various alphabets vs. tasks related to classifying various animals). Several other meta-learning methods, such as [2–10], also face similar limitations. One straightforward approach to overcome this issue is to use MAML and learn a separate initialization per mode. However, this requires knowing the mode of each task (i.e., the groundtruth task-mode-label) both during meta-training and meta-test time, which is often unavailable in real-world scenarios. Additionally, such an approach may prevent the sharing of relevant knowledge across related tasks from distinct but related modes, thus potentially limiting the model's performance.

This paper introduces M3L (MultiModal Meta-Learning), a new meta-learning approach for situations where tasks are sampled from a multimodal distribution with unknown modes. The method extends the MAML framework [1] by jointly learning a good initialization of two complementary neural networks: a base network and a generator network. To enable mode-specific and task-specific adaptation, the base network is modulated with *modulation layers* that apply a “feature-wise gating” mechanism to selectively pass forward some activations while zeroing out others. The generator network predicts the parameters of these modulation layers by taking a few training examples from the given task, producing a task embedding, and subsequently transforming it to generate the modulation parameters required to condition the base network on that task. Notably, these modulation parameters act as parameters in the base network, but are predictions from the generator network.

Recently, a few approaches have been proposed to tackle meta-learning from multimodal task distributions. Some of these methods use a low-dimensional task

representation to transform the model's parameters [11, 12] or generate additional context parameters for the model [13, 14]. Other techniques perform task clustering either in the task space or parameter space [15–18], and learn a separate model initialization for tasks within each cluster. Similarly, the work in [19] learns multiple model initializations but it uses a task encoder network to select the best one to fine-tune for a given task. These approaches can be challenging to train end-to-end and require careful tuning of various hyperparameters, such as the number of context parameters, the number of clusters or model initializations. Unlike existing methods, M3L is not dependent on the number of modes (or datasets) in the task distribution, and allows for joint adaptation of a generator and a base model, resulting in improved performance after only a few adaptation steps at test time. The experimental evaluation demonstrates that M3L learns to modulate the base model's activations effectively, leveraging only relevant information for the given task. This results in a faster learning process and higher prediction accuracy. The results demonstrate that M3L outperforms meta-learning approaches that learn a single initialization across all tasks and existing methods designed for meta-learning from multimodal task distributions.

To summarize, the contributions of this work are as follows:

- We propose a novel meta-learning framework designed for multimodal task distributions with unknown modes, addressing limitations posed by complex scenarios. The proposed approach extends MAML [1] by jointly learning a robust initialization for two complementary neural networks: a base network and a generator network.
- The introduction of modulation layers in the base networks significantly improves mode-specific and task-specific adaptability, allowing for more effective and efficient learning.
- The proposed use of modulation layers facilitates faster learning and higher accuracy after minimal adaptation steps at test time. This showcases superior performance compared to existing multimodal and unimodal meta-learning methods.

2 Background and related work

In recent years, meta-learning has emerged as a powerful paradigm for few-shot learning, enabling a learner to effectively learn unseen tasks with only a few samples by leveraging prior knowledge learned from related tasks. During the meta-training phase, training tasks $\{\mathcal{T}_i\}_{i=1}^T$ are

sampled from a task distribution $P(\mathcal{T})$. Each task \mathcal{T}_i corresponds to data generating distributions $\mathcal{T}_i \triangleq \{p_i(x), p_i(y|x)\}$, and the data sampled from each task is divided into a small *support set* $\mathcal{D}_i^{(sp)}$ containing K training examples and a large *query set* $\mathcal{D}_i^{(qr)}$. During the meta-test phase, a completely new task \mathcal{T}_{new} is sampled from $P(\mathcal{T})$, and a small set $\mathcal{D}_{new}^{(sp)} \triangleq \{x_k, y_k\}_{k=1}^K$ containing a few training examples is observed. The goal is to train a model on $\mathcal{D}_{new}^{(sp)}$ while leveraging the previous knowledge acquired during meta-training in order to achieve good generalization performance on new unlabeled test examples from \mathcal{T}_{new} .

Optimization-based meta-learning methods [1, 9, 10, 20, 21] aim to learn models that can efficiently be adapted, through an optimization procedure, to new tasks with a few training examples. Among these, MAML [1] uses a bi-level optimization process to learn an initial set of parameters θ that can be efficiently fine-tuned with gradient descent on new tasks. In particular, MAML meta-trains a neural network model f_θ parameterized by θ using a two-stage procedure consisting of an inner and an outer loop. In the inner loop, the initial parameters θ are adapted to each training task \mathcal{T}_i by taking a few gradient descent steps on the support set $\mathcal{D}_i^{(sp)}$, resulting in task-specific parameters θ'_i . This is illustrated in Eq. 1 when a single gradient step is used:

$$\theta'_i = \theta - \alpha \nabla_{\theta} \mathcal{L}_{\mathcal{D}_i^{(sp)}}(f_\theta). \quad (1)$$

The initial parameters θ are then optimized in the outer loop by minimizing the loss achieved by task-specific parameters θ'_i on the query set $\mathcal{D}_i^{(qr)}$. This is shown in Eq. 2:

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{\mathcal{T}_i \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{D}_i^{(qr)}}(f_{\theta'_i}). \quad (2)$$

The result is a model initialization θ that can effectively be adapted to new tasks using only a few (K) training examples and a few gradient updates.

While the effectiveness of these methods has been demonstrated, there are concerns about their ability to tackle broader meta-learning challenges. The reason being that a single set of meta-parameters θ , which serves as the model's initialization, may not be sufficient when dealing with a diverse range of tasks drawn from a multimodal distribution $P(\mathcal{T})$. In other words, the model's performance could be limited when faced with a wide variety of tasks that require different meta-parameters for optimal results. To overcome this challenge, various methods [12, 13, 15–19, 22] have been developed to integrate task-specific information into the meta-learning framework. For example, MMAML [12] and its revisited version [22], build upon the standard MAML approach by estimating the

mode of tasks sampled from a multimodal task distribution $P(\mathcal{T})$ and modulating the initial model's parameters accordingly. However, only the initial parameters are adapted, rather than the modulated version, which can slow down the process and reduce the model's effectiveness with an increasing number of modes. Another approach, TSA-MAML [18], combines MAML with a k-means clustering in the parameter space to create multiple model initializations (equal to the number of modes). However, this centroid-based clustering fails to take advantage of negative correlations between tasks (e.g., \mathbf{w} and $-\mathbf{w}$ may be assigned to different clusters) and fails to handle tasks that are distant from all clusters. MUSE [19] meta-learns multiple model initialization and uses a task encoder network to select the initialization that will result in the best performance after adaptation to the given task. CAVIA [13] partitions the initial set of model parameters into parameters that are shared across all tasks, and context parameters that are specific to each task. At meta-test time only the context parameters are adapted to each new task. However, both MUSE and CAVIA require carefully tuning various hyperparameters, such as the number of model initialization or context parameters, and involve an initial offline phase of training to learn knowledge that will be frozen during adaptation.

CAVIA has also inspired related works that aim at building a “universal representation” of robust features that lead to strong performance across multiple datasets in a multi-task learning setup [23]. Building on this, the authors in [24, 25] propose to use meta-learning to specialize the universal representation toward each new task. However, learning such a representation in advance is challenging and may result in overfitting. To overcome these issues, SUR [26] and URL [27] train a separate feature extractor for each dataset (or mode) and combine the learned representations to solve a new task at test time. Similarly, FLUTE [28] aims to learn a shared model across all datasets while allowing for specialization to each individual dataset by learning a small set of dataset-specific parameters. However, these approaches are currently limited to few-shot classification problems, and they do not benefit from meta-learning to adapt fast, i.e., using a few adaptation steps at test time.

Building upon these concepts, we propose M3L, a novel framework that extends MAML to effectively handle multimodal task distributions, where tasks are derived from multiple datasets (corresponding to modes). Despite the similarities to MMAML, we introduce a key distinction by modulating the activations of the network rather than directly adjusting the weights. Furthermore, both the generator and the base network parameters in M3L undergo meta-training, in contrast to MMAML's modulation network which only predicts modulation parameters based on

the input task without meta-learning. Our approach operates under the realistic assumption that the mode of each task is unknown, and that the number of modes is also unknown. Nevertheless, we also explore the possibility of incorporating auxiliary information, such as the task mode identity, to further enhance the performance if such information is available. We compare M3L with related approaches, including MMAML [12] and TSA-MAML [18], and demonstrate its superiority through experiments. It is worth noting that our work deals with multimodality in the task distribution, which is distinct from multimodality in data type [29], where tasks represent the same concept but in different modalities, such as a combination of images and text. Interested readers can find more information about meta-learning and its application to multimodal task distributions in [30].

3 Proposed approach

Our proposed approach involves the joint meta-training of two neural networks: a *base network* f_ψ parameterized by ψ , and a *generator network* g_ρ parameterized by ρ , as shown in Fig. 1b.

To enable efficient adaptation of the base model f_ψ to a new task from any mode, we introduce *modulation layers* into its architecture. The goal of these modulation layers is to “condition” the base network f_ψ on that task, by applying a feature-wise gating mechanism to the activations of the preceding layer (as described in Sect. 3.2), allowing only the features (or activations) that are relevant for the task to propagate forward, while disregarding the non-relevant ones.

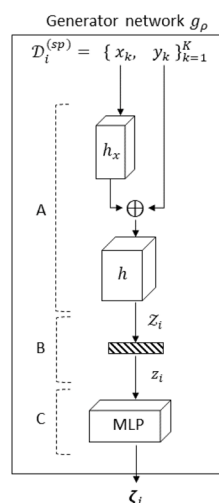
The parameters of the modulation layers are predicted by the generator network g_ρ . To do so, this latter takes the task’s support set as input, produces an embedding or representation of the task, and transforms it to generate the modulation parameters. The task embedding is designed to be invariant to the permutation of examples in the support set, which ensures that the predicted modulation parameters are independent of the ordering of the support set.

The architecture of the generator network is described in Sect. 3.1, and the complete algorithm is presented in Sect. 3.2.

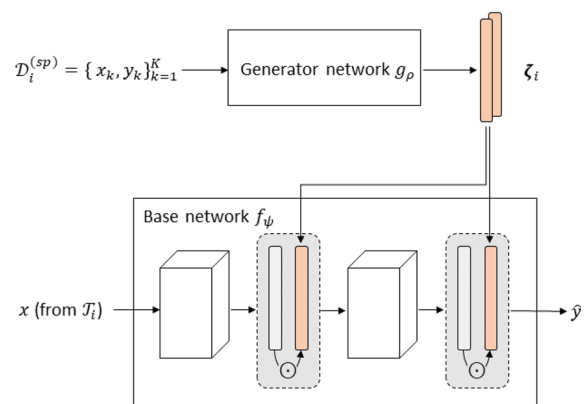
3.1 Generator

As illustrated in Fig. 1a, the generator network g_ρ consists of three parts: a feature extractor (A), an averaging layer (B), and a multilayer perceptron (MLP) (C). The feature extractor (A) takes as input a support set $\mathcal{D}_i^{(sp)} = \{x_k, y_k\}_{k=1}^K$ consisting of inputs x_k and labels y_k (represented as one-hot-vectors in classification) sampled from a task \mathcal{T}_i . It then transforms them into $\mathcal{Z}_i = \{h(h_x(x_k) \oplus y_k)\}_{k=1}^K$, where $h_x(\cdot)$ denotes an input-specific feature extractor, and \oplus denotes the concatenation operator. In other words, it transforms each sample $\mathbf{x}_k = h_x(x_k)$, it concatenates to it the corresponding label and transforms them as $h(\mathbf{x}_k \oplus y_k)$. The averaging layer (B) calculates the mean of the transformed data \mathcal{Z}_i along the K examples, creating a vector representation z_i that characterizes task \mathcal{T}_i . This task representation is invariant to permutations of the examples and is expected to capture task-specific information, such as the task mode. The final part (C) of the network generates the parameters for the modulation layers in the base network. It is an MLP

Fig. 1 Model overview. **a** The generator network consists of three parts: a feature extractor (A), an averaging layer (B), and an MLP (C). **b** The generator takes in input K training examples from a task \mathcal{T}_i (i.e., $\mathcal{D}_i^{(sp)}$) and predicts the parameters of the modulation layers ζ_i . These parameters are used to modulate the activations of the base network. \oplus denotes the concatenation operator and \odot denotes a modulation operator



(a) Generator network architecture



(b) M3L

that takes a task representation z_i as input and predicts a set of parameters $\zeta_i = \{\zeta_i^1, \dots, \zeta_i^L\}$, where L is the number of hidden layers to modulate in the base model. Each ζ_i^l , for $l = 1, \dots, L$, corresponds to the parameters of the modulation layer following the l^{th} layer in the base model. These parameters will act as “scaling” or “gating” parameters; they adjust the feature importance and diminish the contribution of less informative features for the given task \mathcal{T}_i . This is described more formally in the next section.

3.2 Algorithm

Before delving into the technical details, let’s provide an intuitive overview of the proposed algorithm. Imagine the base network f_ψ as a dynamic learner and the generator network g_ρ as a guide that helps the base network adapt quickly to new tasks sampled from different modes. To achieve this, *modulation layers* are added to the base network architecture as described in Sect. 3.1. These layers are predicted by the generator network based on the support set of the task and allows the base network to selectively focus on relevant information while disregarding non-relevant ones. The complete process is presented in Algorithm 1.

Algorithm 1 involves sampling a mini-batch of training tasks from a multimodal distribution $P(\mathcal{T})$, and for each task \mathcal{T}_i , a support dataset $\mathcal{D}_i^{(sp)}$ and a query dataset $\mathcal{D}_i^{(qr)}$ are sampled. In line 6 of Algorithm 1, the *Modulate & Adapt* function (Algorithm 2) is called to predict the modulation parameters and adapt the parameters ρ and ψ to the task at hand.

In line 3 of Algorithm 2, the support set $\mathcal{D}_i^{(sp)}$ is input into the generator g_ρ to obtain the modulation parameters ζ_i , as described in Sect. 3.1. These parameters are then

used to modulate the activations of the base network, resulting in a *modulated model* denoted by $f_\psi|\zeta_i$ which is more specialized for the task at hand. To better explain the modulated network $f_\psi|\zeta_i$, let us examine the activations $a(\mathbf{x}|\psi^l)$ produced by a particular layer l . Using modulation parameters ζ_i^l , it is possible to modulate these activations and get modulated activations denoted as $a(\mathbf{x}|\psi^l) \odot \zeta_i^l$, as exemplified by Eq. 3. This equation outlines a “sigmoidal gating” mechanism that enables ζ_i^l to determine which activations are propagated forward to the next layers and which are zeroed out. It is worth noting that other ways to modulate the activations are also possible, as shown later in Sect. 4.1.

$$a(\mathbf{x}|\psi^l) \odot \zeta_i^l = a(\mathbf{x}|\psi^l) \otimes \sigma(\zeta_i^l), \tag{3}$$

where \otimes denotes an element-wise multiplication and σ is the sigmoid function making each value of ζ_i^l in the range $[0, 1]$. The parameters ρ and ψ are then adapted to the specific task resulting in task-specific parameters ρ' and ψ' , respectively (lines 4 and 5 of Algorithm 2). This adaptation is performed as in the standard framework (see Eq. 1), but the loss $\mathcal{L}_{\mathcal{D}_i^{(sp)}}$ is now computed after transforming the activations of the base model with the parameters of the modulation layers, i.e., using $f_\psi|\zeta_i$. This process can be iterated for $Q \geq 1$ gradient descent steps and the adapted parameters ψ'_i and ζ_i are returned.

In lines 8 and 9 of Algorithm 1, the initial set of parameters ρ and ψ are updated. To do so, the post-adaptation loss is computed with $f_{\psi'_i}|\zeta_i$ on the query set $\mathcal{D}_i^{(qr)}$ and used to update the initial parameters ρ and ψ . Here, any optimizer of choice, e.g., Adam, can be used (not necessarily gradient descent).

Algorithm 1 M3L

Require: Task distribution $P(\mathcal{T})$, step sizes α, β

- 1: randomly initialize ρ and ψ
- 2: **while** not done **do**
- 3: Sample a mini-batch of tasks $\mathcal{T}_i \sim P(\mathcal{T})$
- 4: **for all** \mathcal{T}_i **do**
- 5: Sample data $\mathcal{D}_i = \mathcal{D}_i^{(sp)} \cup \mathcal{D}_i^{(qr)}$ from \mathcal{T}_i
- 6: $\psi'_i, \zeta_i \leftarrow \text{Modulate \& Adapt}(Q, \mathcal{D}_i^{(sp)}, \rho, \psi)$
- 7: **end for**
- 8: Update $\rho \leftarrow \rho - \beta \nabla_\rho \sum_{\mathcal{T}_i \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{D}_i^{(qr)}}(f_{\psi'_i}|\zeta_i)$
- 9: Update $\psi \leftarrow \psi - \beta \nabla_\psi \sum_{\mathcal{T}_i \sim P(\mathcal{T})} \mathcal{L}_{\mathcal{D}_i^{(qr)}}(f_{\psi'_i}|\zeta_i)$
- 10: **end while**
- 11: **return** ρ, ψ

Algorithm 2 *Modulate & Adapt*($Q, \mathcal{D}^{(sp)}, \rho, \psi$)

Require: Adaptation steps Q , support set $\mathcal{D}^{(sp)}$, initial parameters of the generator ρ , initial parameters of the base model ψ

- 1: Initialize $\rho' \leftarrow \rho, \psi' \leftarrow \psi$
- 2: **for** $q = 1, \dots, Q$ **do**
- 3: Predict modulation parameters $\zeta = g_{\rho'}(\mathcal{D}^{(sp)})$
- 4: $\rho' = \rho' - \alpha \nabla_{\rho'} \mathcal{L}_{\mathcal{D}^{(sp)}}(f_{\psi'} | \zeta)$
- 5: $\psi' = \psi' - \alpha \nabla_{\psi'} \mathcal{L}_{\mathcal{D}^{(sp)}}(f_{\psi'} | \zeta)$
- 6: **end for**
- 7: Predict modulation parameters $\zeta = g_{\rho'}(\mathcal{D}^{(sp)})$
- 8: **return** ψ', ζ

At the end of meta-training, the parameters ρ and ψ are returned (line 11) and act as initializations of the two networks that can be effectively adapted to new tasks. Indeed, at meta-test time, when a completely new task \mathcal{T}_{new} is sampled from $P(\mathcal{T})$, only its support set $\mathcal{D}_{\text{new}}^{(sp)}$ containing a few training examples is observed. The goal is to efficiently adapt the initial set of parameters ρ and ψ using $\mathcal{D}_{\text{new}}^{(sp)}$, as outlined in Algorithm 2, by making the generator produce appropriate task-specific modulation parameters ζ_{new} for the base network. The adapted model, together with the modulation layers, can then be used to make accurate predictions on unseen input data from \mathcal{T}_{new} .

3.3 Differences with previous approaches

Though our proposed approach shares some similarities with MMAML, it is important to clarify the key distinctions between the two approaches.

While both M3L and MMAML aim to handle multimodal task distributions, they adopt different strategies to adapt the base model to new tasks. In MMAML, modulation is achieved through direct adjustments to the weights of the base network using the output of the modulation network, as $\psi^l \odot \zeta_i^l$. In contrast, our approach introduces modulation layers into the base network's architecture. These modulation layers condition the base network by acting on the activation of the preceding layer, as shown in Eq. 3. This allows our method to selectively activate relevant features for each task while suppressing non-relevant ones, effectively modulating the network's behavior without directly modulating its weights. Another fundamental difference lies in the meta-training of the generator network. In MMAML, the parameters of the modulation network, unlike those of the base network, are not subject to meta-learning. Instead, they are learned during training and kept fixed during meta-testing. The modulation network solely focuses on predicting modulation parameters

based on the support set of the input task. Therefore, MMAML does not actively seek an initialization of the modulation network parameters for easy adaptability to new tasks. Its primary emphasis remains on predicting task-specific modulation parameters for the current task without explicitly optimizing for broader generalization across tasks. In contrast, both the generator and the base network in M3L are meta-trained. This joint meta-training process allows the generator to better capture task-specific information and produce more effective modulation parameters, leading to improved performance in handling diverse tasks.

4 Experiments

In this section, M3L is evaluated using tasks from various few-shot regression and image classification datasets. The selection of the baseline methods is performed to ensure a comprehensive evaluation of M3L, and it aligns with related works, such as [12, 18, 19], ensuring consistency with existing research. In particular, results are compared against six different baselines:

- “Scratch”: A naive approach that consists of training a model on each new task from scratch, i.e., with a random parameters' initialization instead of meta-learning it. This baseline is used as a lower bound on the performance.
- MAML [1] and Reptile [9]: Two widely used meta-learning approaches.
- MMAML [12] and TSA-MAML [18]: Two established methods for meta-learning from multimodal task distributions.
- Multi-MAML: A straightforward extension of MAML for multimodal task distributions, assuming that the mode of each task is known. In Multi-MAML, a separate model is meta-trained for tasks within each

separate mode using MAML. At meta-test time, the mode of each new task is assumed to be known and used to select the corresponding initial model to be adapted for the new task. Note that directly comparing the other approaches to Multi-MAML is not fair as this latter uses additional information (mode label for each task) that is usually unavailable in real-world situations. However, this baseline can provide useful insights into whether or not it is helpful to transfer knowledge across different modes. In the result tables, we denote this baseline as “Multi-MAML (tml)” to indicate that it uses additional information (i.e., task-mode-label) not available for the other methods.

To ensure fair comparisons, similar hyperparameters and model architectures are used for all methods. The selection is performed through a tuning process aimed at finding a set of parameters that works well for all approaches. Additionally, in TSA-MAML, the number of clusters is set equal to the number of modes to prevent affecting its performance. All methods are evaluated using 100 test tasks from each mode (or dataset) and the average performance is reported after fine-tuning the models for 20 adaptation steps. The final results are the average and the standard deviation over four complete runs of the algorithms, including meta-training and meta-testing. All experiments detailed in the paper are executed on a single Nvidia A100-SXM4 GPU with 40GB of RAM using Python and the PyTorch library.

4.1 Regression results

We start the experimental evaluation with a simple multimodal few-shot regression problem. The multimodal task distribution is constructed considering five different families of functions from which tasks are generated. These families are: (1) sinusoidal $y(x) = a \sin(x - b)$, where

$a \sim U[0.1, 5.0]$, $b \sim U[0, \pi]$; (2) linear $y(x) = ax + b$, where $a \sim U[0, 1]$, $b \sim U[0, 5]$; (3) quadratic $y(x) = ax^2 + bx + c$, where $a, b, c \sim U[0, 0.5]$; (4) 11 norm $y(x) = a|x - c| + b$, where $a, b, c \sim U[0, 0.5]$; (5) hyperbolic tangent $y(x) = a \tanh(x - c) + b$, where $a, b, c \sim U[0, 0.5]$. Each task is randomly sampled from one of the five underlying families and consists of inputs x sampled uniformly in $[-5, 5]$. The base model and the generator model consist of two hidden layers, with sizes 25 and 50, followed by batch normalization and ReLU nonlinearities. The adaptation phase (line 6 of Algorithm 1) consists of $Q = 2$ gradient descent steps with a fixed learning rate $\alpha = 0.005$, while the update of the models' parameters (lines 8–9 in Algorithm 1) is performed using Adam optimizer with $\beta = 0.001$.

Results are reported in Table 1 for 5-shot ($K = 5$) and 10-shot ($K = 10$) learning problems. As expected, MAML and Reptile exhibit high errors when tasks are sampled from a multimodal distribution, as they attempt to learn a single initialization of neural network parameters adaptable for each task, regardless of its mode. Learning such an initialization for tasks sampled from different modes, as in this case, is challenging and may not be sufficient to obtain good post-adaptation performance at test time. Also, TSA-MAML shows poor performance in this regression setting, likely due to the limitations related to directly applying k-means clustering in the parameters space and the fact that parameters specific to various tasks might not constitute clearly separable clusters (e.g., when tasks from various modes are similar or related). The proposed approach achieves a good performance (low MSE) for both $K = 5$ -shot and $K = 10$ -shot learning scenarios. Notably, for $K = 5$, M3L outperforms MMAML by a significant margin, indicating its superiority in handling multimodal task distributions. This result suggests that the feature-wise gating mechanism in M3L that modulates the activations of the base model is better than the direct modulation of the model's parameters used in MMAML. It also shows that by jointly adapting both the generator model and the base model at test time, M3L enables an effective adaptation to a diverse range of tasks. Finally, despite not having access to task-mode labels, M3L outperforms the Multi-MAML baseline in the $K = 10$ -shot learning scenario, demonstrating the importance of sharing knowledge across related modes. This is supported by the visual evidence in Fig. 2a, which shows that the parameters of the last modulation layer share a common structure across different modes, providing an advantage for transferring knowledge across modes.

In addition to the feature-wise gating defined in Eq. 3, we also investigated an alternative way to modulate the activations in our method by using an affine transformation, as used in FiLM [31], where the modulation parameters ζ^l of a layer l consist of two vectors, ζ_a^l and ζ_b^l , used to scale and bias the activations, i.e.,

Table 1 Regression results (average mean-squared-error MSE) using tasks sampled from 5 (sinusoidal, linear, quadratic, 11norm, hyperbolic tangent) modes. The results are reported in the 5-shot ($K = 5$) and 10-shot ($K = 10$) learning scenarios after $Q = 20$ adaptation steps at test time

Method	5-shot	10-shot
Scratch	1.80 ± 0.31	1.07 ± 0.12
Reptile	1.02 ± 0.19	0.33 ± 0.07
MAML	0.86 ± 0.18	0.36 ± 0.07
MMAML	0.58 ± 0.35	0.08 ± 0.03
TSA-MAML	1.92 ± 0.54	0.66 ± 0.11
Multi-MAML (tml)	0.16 ± 0.03	0.10 ± 0.07
M3L	0.21 ± 0.07	0.06 ± 0.05

The best results are indicated with a bold font

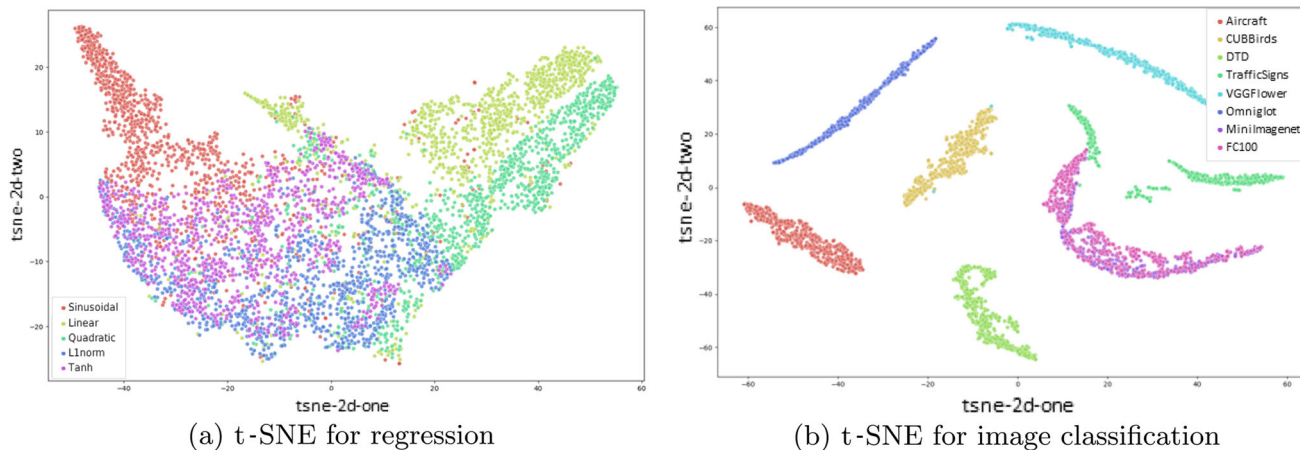


Fig. 2 Visualization of the modulation parameters generated by g_ρ for the last modulation layer L (i.e., ζ^L). The parameters are visualized in the regression setting (a) and the image classification setting (b). The

plots are created after applying PCA with 25 components [38] and t-SNE [39] with perplexity equal to 25

$$a(\mathbf{x} \psi^l) \odot \zeta^l = \zeta_a^l \otimes a(\mathbf{x} \psi^l) + \zeta_b^l.$$

With this modulation, M3L achieves an average MSE of 0.28 ± 0.08 instead of 0.21 ± 0.07 in the 5-shot scenario, and 0.10 ± 0.06 instead of 0.06 ± 0.05 in the 10-shot scenario. This is likely because gating the model’s activations with a value between 0 and 1 is sufficient to specialize the model to specific tasks while simultaneously leveraging, to various degrees, the representations learned from other related tasks. Therefore, in the rest of the experiments, all results are reported using a feature-wise gating mechanism.

4.2 Classification results

In the classification setting, a task is defined by randomly selecting N classes and K labeled images per class from a given dataset, i.e., N -way K -shot classification problem. To create a multimodal task distribution, multiple well-

established datasets are combined together, each one representing a different mode. The datasets include Mini-ImageNet [3], FC100 [8], Omniglot [32], Aircraft [33], CUB Birds [34], Describable Textures Dataset (DTD) [35], Traffic Signs [36], and VGG Flowers [37]. The classes within each dataset are split into two sets, one used to generate tasks for meta-training and the other to generate tasks for meta-testing. This is performed by following the train/test splits of [3] for Mini-ImageNet, [8] for FC100, and [20] for the remaining ones. Images sampled from these datasets are converted to RGB format with a resolution of 32×32 pixels. The base model is composed of three modules, each consisting of a convolutional layer with 64 filters, followed by batch normalization and ReLU nonlinearities. Additionally, three linear layers with a size of 576 are used to complete the classification model. Similarly, the architecture of the generator consists of the same three modules followed by four linear layers with sizes of 100, 100, 200, 200, respectively. The adaptation

Table 2 Classification results (average accuracy) when tasks are sampled from a multimodal task distribution consisting of MiniImagenet, FC100, Omniglot, Aircraft, CUB Birds, DTD, Traffic Signs, VGG Flowers. The columns “AllDatasets” report the average

performance when tasks are randomly sampled from all datasets. The results are reported in the 5-way ($N = 5$) 5-shot ($K = 5$) learning scenarios after $Q = 20$ adaptation steps at test time

Method	Mini	FC100	Omni	Aircraft	CUB	DTD	Traffic	VGG	AllDatasets
Scratch	0.38	0.49	0.90	0.39	0.40	0.32	0.58	0.66	0.52 ± 0.02
Reptile	0.45	0.60	0.91	0.54	0.46	0.40	0.77	0.66	0.59 ± 0.01
MAML	0.44	0.56	0.96	0.62	0.49	0.33	0.66	0.64	0.61 ± 0.03
MMAML	0.49	0.62	0.97	0.64	0.59	0.33	0.81	0.69	0.63 ± 0.01
TSA-MAML	0.41	0.52	0.96	0.61	0.52	0.32	0.68	0.64	0.58 ± 0.01
Multi-MAML (tml)	0.37	0.54	0.97	0.72	0.57	0.33	0.85	0.70	0.61 ± 0.03
M3L	0.49	0.63	0.97	0.65	0.54	0.33	0.77	0.70	0.66 ± 0.02

The best results are indicated with a bold font

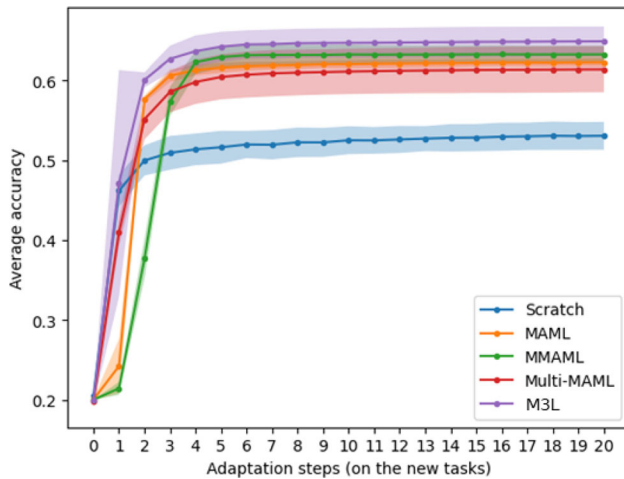


Fig. 3 Average accuracy and standard deviation for different adaptation steps at test time, when test tasks are randomly sampled from the multimodal task distribution. In the plot, results are reported for the following methods: “Scratch”, MAML, MMAML, Multi-MAML, and M3L. The x-axis represents the number of adaptation steps, and the y-axis represents the average accuracy over multiple test tasks. The shaded regions around the curves represent the standard deviation of the accuracy

phase (line 6 of Algorithm 1) consists of $Q = 2$ gradient descent steps with $\alpha = 0.008$ and the models’ parameters are updated with Adam optimizer and $\beta = 0.001$ (in line 8–9 of Algorithm 1).

Results are reported in Table 2. Overall, when test tasks are randomly sampled from all datasets (column “AllDatasets” in the table), M3L outperforms other meta-learning methods. In particular, it outperforms the Multi-MAML baseline by a good margin (i.e., 5%), highlighting the importance of sharing knowledge from related modes (or datasets) to make accurate predictions. This idea is reinforced by Fig. 2b, which shows the parameters of the last modulation layers predicted by the generator. While the generator generates distinct modulation parameters for tasks from different datasets (e.g., Aircraft, CUB Birds, DTD, Traffic Signs, VGG Flowers, Omniglot), it also generates similar modulation parameters for tasks within related datasets such as MiniImageNet and FC100, enabling the transfer of information across tasks sampled from these datasets. Indeed, MiniImageNet and FC100 have some commonalities, such as similar types of images and some classes in common, that enable M3L to modulate the base model using information from both datasets. Moreover, by modulating the activations of the base model, adaptation to a new task requires only a few gradient steps to achieve good performance. This is demonstrated in Fig. 3, where the average accuracy of M3L increases rapidly and reaches high performance after only 2 adaptation steps. This is different from the MMAML approach,

Table 3 Regression and classification results (average MSE and average accuracy respectively) when task-mode-labels are incorporated in the proposed approach, i.e., M3L (tml). The results are reported for 5-shot learning ($K = 5$), randomly sampling tasks from all modes (or datasets) in the multimodal task distribution

Method	Regression	Classification
Scratch	1.80 ± 0.31	0.52 ± 0.02
Reptile	1.02 ± 0.19	0.59 ± 0.01
MAML	0.86 ± 0.18	0.61 ± 0.02
MMAML	0.58 ± 0.35	0.63 ± 0.01
TSA-MAML	1.92 ± 0.54	0.58 ± 0.01
Multi-MAML (tml)	0.16 ± 0.03	0.61 ± 0.03
M3L	0.21 ± 0.07	0.66 ± 0.02
M3L (tml)	0.07 ± 0.02	0.65 ± 0.01

The best results are indicated with a bold font

which does not adapt the parameters used for the modulation at test time, thereby slowing the adaptation process.

4.3 Additional results

Although M3L is designed to address the realistic setting where the mode from which a task is sampled is unknown, we conducted an additional experiment to explore the scenario where auxiliary information, such as the task-mode-label (tml), is available to our method, similar to the setting in Multi-MAML. In this case, we propose to enhance the generator network’s predictions of the modulation parameters for each task \mathcal{T}_i by incorporating this extra information. We achieve this by appending a one-hot-vector representation of the task-mode-label to the task embedding z_i before the last part of the generator network (see Fig. 1b for reference). By doing so, we enable the final MLP to leverage both the task embedding and the task-mode-label to predict the modulation parameters, which can result in improved performance. The modulation parameters predicted by the generator network for each task now depend not only on the support set of the task but also on the actual mode (or dataset) from which the task is sampled. Our results (Table 3) indicate that incorporating auxiliary information as an input to the generator is highly advantageous for the regression problems, leading to further improvement in the performance (i.e., a decrease in the average MSE from 0.21 to 0.07). However, classification results remain almost the same. This is most likely due to the fact that the mode of a task is readily identifiable from the task data in the classification problems, obviating the need for incorporating the task-mode-label. In contrast, the representations learned for tasks sampled from different modes are more similar (or less distinguishable) in the

regression problems, as confirmed by the t-SNE plot in Fig. 2a. Therefore, unlike Multi-MAML, incorporating auxiliary information (if available) to the generator in our proposed way, can help distinguish different modes and simultaneously leverage the knowledge learned from tasks in related modes, leading to improved performance.

5 Conclusion and future work

In this work, we proposed a novel approach that enhances model-agnostic meta-learning with feature-based modulations to effectively handle multimodal task distributions. Our method utilizes two jointly trained neural networks to enhance and expedite model adaptation to tasks sampled from any mode. Specifically, a generator network learns to embed a target task and predicts parameters for the modulation layers of the base model, enabling the model to effectively specialize to the target task and improve performance. Our approach shows promising experimental results in the challenging setting of multimodal task distributions, outperforming existing meta-learning methods, including those designed for multimodal task distributions. Our results highlight the importance of leveraging “relevant” past experience to achieve accurate predictions on new tasks. We also explored the potential benefit of incorporating auxiliary information, such as the task-mode-label, to our method. Our experiments demonstrated that incorporating such information can further enhance the performance of our method, if available. As future work, it would be interesting to investigate the extension of our approach to more complex multimodal task distributions, and to explore the use of other types of auxiliary information that may be available for different applications.

Acknowledgements This work was supported by the “Knowledge Foundation” (KK-stiftelsen).

Author contributions Anna Vettoruzzo, Mohamed-Rafik Bouguelia, and Thorsteinn Rögnvaldsson contributed to conceptualization; Anna Vettoruzzo and Mohamed-Rafik Bouguelia involved in methodology; Anna Vettoruzzo involved in formal analysis and investigation; Anna Vettoruzzo involved in writing—original draft preparation; Mohamed-Rafik Bouguelia involved in writing—review and editing; Mohamed-Rafik Bouguelia and Thorsteinn Rögnvaldsson involved in supervision.

Funding Open access funding provided by Halmstad University. “Knowledge Foundation” (KK-stiftelsen).

Availability of data and materials All datasets used in the experiments are open-source datasets available online.

Code availability The code will be published online after acceptance of the manuscript.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. PMLR, pp 1126–1135
2. Vinyals O, Blundell C, Lillicrap T, Wierstra D et al (2016) Matching networks for one shot learning. *Adv Neural Inf Process Syst* 29
3. Ravi S, Larochelle H (2017) Optimization as a model for few-shot learning. In: International conference on learning representations
4. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. *Adv Neural Inf Process Syst* 30
5. Mishra N, Rohaninejad M, Chen X, Abbeel P (2017) A simple neural attentive meta-learner. In: International conference on learning representations (ICLR)
6. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: relation network for few-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1199–1208
7. Rusu AA, Rao D, Sygnowski J, Vinyals O, Pascanu R, Osindero S, Hadsell R (2018) Meta-learning with latent embedding optimization. In: International conference on learning representations
8. Oreshkin B, Rodríguez López P, Lacoste A (2018) Tadam: task dependent adaptive metric for improved few-shot learning. *Adv Neural Inf Process Syst* 31
9. Nichol A, Achiam J, Schulman J (2018) On first-order meta-learning algorithms. Preprint [arXiv:1803.02999](https://arxiv.org/abs/1803.02999)
10. Rajeswaran A, Finn C, Kakade SM, Levine S (2019) Meta-learning with implicit gradients. *Adv Neural Inf Process Syst* 32
11. Garnelo M, Rosenbaum D, Maddison C, Ramalho T, Saxton D, Shanahan M, Teh YW, Rezende D, Eslami SA (2018) Conditional neural processes. In: International conference on machine learning. PMLR, pp 1704–1713
12. Vuorio R, Sun S-H, Hu H, Lim JJ (2019) Multimodal model-agnostic meta-learning via task-aware modulation. *Adv Neural Inf Process Syst* 32

13. Zintgraf L, Shiarli K, Kurin V, Hofmann K, Whiteson S (2019) Fast context adaptation via meta-learning. In: International conference on machine learning. PMLR, pp 7693–7702
14. Li H, Dong W, Mei X, Ma C, Huang F, Hu B-G (2019) LGM-Net: learning to generate matching networks for few-shot learning. In: International conference on machine learning. PMLR, pp 3825–3834
15. Yao H, Wei Y, Huang J, Li Z (2019) Hierarchically structured meta-learning. In: International conference on machine learning. PMLR, pp 7045–7054
16. Jiang W, Kwok J, Zhang Y (2022) Subspace learning for effective meta-learning. In: International conference on machine learning. PMLR, pp 10177–10194
17. Jerfel G, Grant E, Griffiths T, Heller KA (2019) Reconciling meta-learning and continual learning with online mixtures of tasks. *Adv Neural Inf Process Syst* 32
18. Zhou P, Zou Y, Yuan X-T, Feng J, Xiong C, Hoi S (2021) Task similarity aware meta-learning: Theory-inspired improvement on MAML. In: Uncertainty in artificial intelligence. PMLR, pp 23–33
19. Vettoruzzo A, Bouguelia M-R, Rögnvaldsson T (2023) Meta-learning from multimodal task distributions using multiple sets of meta-parameters. In: 2023 international joint conference on neural networks (IJCNN). IEEE, pp 1–8
20. Triantafillou E, Zhu T, Dumoulin V, Lamblin P, Evci U, Xu K, Goroshin R, Gelada C, Swersky K, Manzagol P-A (2019) Meta-dataset: a dataset of datasets for learning to learn from few examples. In: International conference on learning representations
21. Raghu A, Raghu M, Bengio S, Vinyals O (2019) Rapid learning or feature reuse? Towards understanding the effectiveness of MAML. In: International conference on learning representations
22. Abdollahzadeh M, Malekzadeh T, Cheung N-MM (2021) Revisit multimodal meta-learning through the lens of multi-task learning. *Adv Neural Inf Process Syst* 34:14632–14644
23. Bilen H, Vedaldi A (2017) Universal representations: the missing link between faces, text, planktons, and cat breeds. Preprint [arXiv:1701.07275](https://arxiv.org/abs/1701.07275)
24. Requeima J, Gordon J, Bronskill J, Nowozin S, Turner RE (2019) Fast and flexible multi-task classification using conditional neural adaptive processes. *Adv Neural Inf Process Syst* 32
25. Liu L, Hamilton WL, Long G, Jiang J, Larochelle H (2020) A universal representation transformer layer for few-shot image classification. In: International conference on learning representations
26. Dvornik N, Schmid C, Mairal J (2020) Selecting relevant features from a multi-domain representation for few-shot classification. In: Computer vision–ECCV 2020: 16th European conference, Glasgow, Proceedings, Part X 16. Springer, pp 769–786
27. Li W-H, Liu X, Bilen H (2021) Universal representation learning from multiple domains for few-shot classification. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 9526–9535
28. Triantafillou E, Larochelle H, Zemel R, Dumoulin V (2021) Learning a universal template for few-shot dataset generalization. In: International conference on machine learning. PMLR, pp 10424–10433
29. Ma Y, Zhao S, Wang W, Li Y, King I (2022) Multimodality in meta-learning: a comprehensive survey. *Knowl-Based Syst* 108976
30. Vettoruzzo A, Bouguelia M-R, Vanschoren J, Rögnvaldsson T, Santosh K (2024) Advances and challenges in meta-learning: a technical review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*
31. Perez E, Strub F, De Vries H, Dumoulin V, Courville A (2018) Film: visual reasoning with a general conditioning layer. In: Proceedings of the AAAI conference on artificial intelligence, vol 32
32. Lake B, Salakhutdinov R, Gross J, Tenenbaum J (2011) One shot learning of simple visual concepts. In: Proceedings of the annual meeting of the cognitive science society, vol 33
33. Maji S, Rahtu E, Kannala J, Blaschko M, Vedaldi A (2013) Fine-grained visual classification of aircraft. Preprint [arXiv:1306.5151](https://arxiv.org/abs/1306.5151)
34. Wah C, Branson S, Welinder P, Perona P, Belongie S (2011) The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology
35. Cimpoi M, Maji S, Kokkinos I, Mohamed S, Vedaldi A (2014) Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3606–3613
36. Stallkamp J, Schlipsing M, Salmen J, Igel C (2012) Man versus computer: benchmarking machine learning algorithms for traffic sign recognition. *Neural netw* 32:323–332
37. Nilsback M-E, Zisserman A (2008) Automated flower classification over a large number of classes. In: 2008 sixth Indian conference on computer vision, graphics & image processing. IEEE, pp 722–729
38. Pearson K (1901) Liii. on lines and planes of closest fit to systems of points in space. *Lond Edinb Dublin Philos Mag J Sci* 2(11):559–572. <https://doi.org/10.1080/14786440109462720>
39. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *J Mach Learn Res* 9(11)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.