The Journal of Supercomputing (2025) 81:1281
https://doi.org/10.1007/s11227-025-07757-y

Unsupervised tractive momentum: a novel unsupervised few-shot learning framework

Zhong Cao[1] · Jiang Lu[2] · Liu He[3] · Yuheng Luo[4]

Received: 19 May 2025 / Accepted: 7 August 2025 / Published online: 28 August 2025

Abstract

Few-shot learning (FSL) aims at distilling transferable knowledge on existing concepts to cope with novel concepts for which only a few labeled data are available. Most of the popular FSL methods acquire this knowledge by learning on large-scale supervised data from the existing concepts. Considering obtaining supervised data might sometimes be difficult and heavy-burden, we pursue a relatively mild prerequisite for FSL, that is, using unsupervised instead of supervised data to acquire the transferable knowledge. We propose a novel easy-to-implement FSL framework, **U**nsupervised **T**ractive **M**omentum (UTM), composed of modular dual encoders, a combinatorial loss mechanism, and a classifier that together form a reusable and extensible learning system, that only requires unsupervised data of existing concepts. UTM randomly samples unsupervised data and augments them to create many synthetic *query-key* matching tasks on-the-fly, and deploys two different encoders while possessing identical architecture, named *traction encoder* and *momentum encoder*, to learn a representation space by a combinatorial parameter updating manner. The representation space learned on unsupervised data is expected to be a good fit to few-shot recognition on novel concepts. UTM is composed of parallelizable dual encoders and optimized for scalable training in GPU-based high-performance computing environments. Theoretical convergence and bound analysis further support its deployment in distributed systems. Theoretical justifications of the parameter updating mechanism in UTM are given from the perspective of convergence, and a theoretical loss bound for UTM is proved, which mathematically quantifies the relationship between our self-supervised UTM and the vanilla supervised method. Extensive experimental evaluation on several benchmark datasets demonstrates that UTM yields significant improvement to state-of-the-art unsupervised methods even very close to supervised methods, which can also be well explained using our theory.

Zhong Cao and Jiang Lu have contributed equally to this work.

Extended author information available on the last page of the article

Keywords Self-supervised learning · Unsupervised learning · Few-shot learning · Momentum update · Bound analysis

1 Introduction

In the past decade, artificial intelligence techniques represented by deep learning [1] have scored great achievements in a broad spectrum of research fields including language [2], vision [3], and speech [4], but they usually entails massive supervised training data. Comparatively speaking, one impressive hallmark of human is the ability of learning and generalizing from very few samples, which is widely considered as one of the noticeable demarcations separating artificial intelligence and human intelligence, since humans can readily establish their cognition to novel concept from just a single or a handful of examples [5–7], whereas machine learning algorithms typically require hundreds or thousands of supervised samples to guarantee generalization. However, many realistic application scenarios do not allow us access to sufficient labeled training data due to some factors including privacy, security or high labeling costs for data, etc. Therefore, few-shot learning (FSL) becomes an eagerly-awaited goal pursued by many machine learning researchers recently [8–22], which is also usually regarded as a necessary trip to develop universal artificial intelligence [23].

In the typical N-way K-shot setting of FSL problems, one support dataset $\{(x_i^s, y_i^s)\}_{i=1}^{NK}$ is given, which corresponds to N novel concepts with only K labeled data per concept (K is very small), and the goal is to correctly classify future query data x^q into one of the N novel concepts. In order to acquire some transferable knowledge conducive to classification on novel concepts, ones are encouraged to capitalize on an auxiliary dataset $\mathcal{A} = \{(x_i, y_i)\}_{i=1}^{n}$ corresponding to some existing concepts with sufficient supervised data per concept (the label space of y^s and y do not overlap). Many FSL methods create their model on this supervised $\mathcal{A}$ and then transfer it to the target task on novel concepts. Unfortunately, the large-scale supervised $\mathcal{A}$ not only burdens human with heavy manual labor on collecting and labeling it but also places limits on the popularization and use of the currently popular FSL methods that build on supervised auxiliary data.

Thus, we pursue a relatively mild condition for performing FSL. Apparently, collecting an unlabeled auxiliary set $\mathcal{A} = \{x_i\}_{i=1}^{n}$ corresponding to some previously seen concepts is more easy-to-implement (e.g., one can readily acquire a large number of unsupervised images via web crawler in the big data era), even if it is still a requirement that the data in the unsupervised auxiliary set are drawn from the same distribution as the data of novel concepts to be classified (i.e., no domain gap exists).

We propose a novel FSL method, **U**nsupervised **T**ractive **M**omentum (UTM), that conducts representation learning on an unsupervised auxiliary set and performs few-shot classification for novel concepts in the learned representation space. The representation learning process of UTM is illustrated in Fig. 1(a). Given a large-scale unsupervised auxiliary set $\mathcal{A}$, UTM randomly samples some unsupervised data and augments them to create many *query-key* matching tasks on-the-fly. Then, two encoders called *traction encoder* and *momentum encoder* that possess the identical

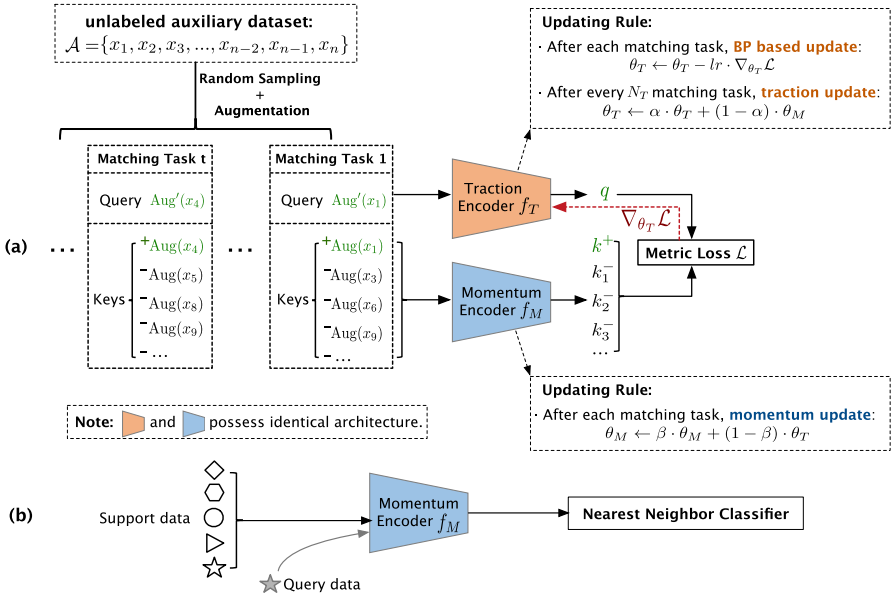

Fig. 1 Framework of UTM. **a** Phase of self-supervised representation learning. Aug(x) and Aug′(x), considered from one same concept, are two different synthetic data randomly augmented from x. **b** Phase of few-shot classification

architecture but different parameters are used for extracting the representation of query and keys, respectively, followed by a task-specific metric loss. Importantly, the two parameterized encoders are updated with different rules: (1) Traction encoder is updated via gradient-based back propagation (BP) after tackling each matching task, and a traction update is performed on it after every N_T matching tasks to avoid excessive deviation between the two encoders. (2) Momentum encoder always moves toward the traction encoder in a momentum manner after each matching task. From a high-level perspective, the traction encoder, mainly driven by the task-level metric loss, works as one engine that offers traction to the momentum encoder.

During the phase of few-shot classification, as shown in Fig. 1(b), the momentum encoder that is well trained on the unsupervised dataset is used for representing the support data and query data in the learned representation space, where the representations belonging to the same concept are expected to cluster close but those belonging to the different concept far apart so that query representation can be classified by a simple nearest neighbor classifier.

In addition to the above, we give a convergence analysis to justify the combinatorial parameter updating mechanism between the two encoders of UTM, and also prove a theoretical loss bound for UTM, which mathematically quantifies the relationship between our unsupervised UTM and the vanilla supervised method. These theoretical analysis can explain our final experimental results well. Experimental evaluation on two popular FSL benchmark datasets, Omniglot and *mini*ImageNet, demonstrates that UTM not only yields significant performance improvement to state-of-the-art unsupervised methods, but also achieves appealing results that are

very close to supervised methods. For instance, our unsupervised UTM reaches the accuracy 98.09% for 5-way 5-shot classification on Omniglot, which is quite competitive compared to 98.83% achieved by supervised Model-agnostic meta-learning method (MAML) [10]. Our method is effective for unsupervised few-shot learning and achieves competitive performance. UTM is built with modular dual encoders that can be trained asynchronously, enabling scalable deployment on high-performance GPU clusters. This structure also allows real-time pretraining on large unlabeled datasets in parallel.

Overall, our main contribution is thus fourfold:

(1) A novel unsupervised few-shot learning framework, UTM, is proposed. This framework deploys two different encoders, traction encoder and momentum encoder, and adopts a combinatorial parameter updating strategy, which can lead to a good feature representation space for few-shot learning.
(2) A convergence analysis is provided to justify the combinatorial parameter updating strategy in our UTM.
(3) A theoretical loss bound for UTM is successfully derived. This bound mathematically quantifies the performance gap between our unsupervised UTM and the vanilla supervised method, and it also clarifies the factors affecting this gap.
(4) An extensive experimental evaluation on various benchmark datasets of image recognition shows that our UTM obtains the state-of-the-art performance for most N-way K-shot settings and even competitive performance compared with some supervised FSL methods.

The rest of this paper is organized as follows. We review related work in Section 2. The proposed UTM framework and the convergence analysis are detailed in Section 3. In Section 4, we comprehensively analyze the theoretical loss bound of UTM. We report the experimental results in Section 5 and conclude our work in Section 6.

2 Related work

2.1 Supervised methods for FSL

Compared with common machine learning paradigm that involves large-scale labeled training data, the development of FSL is tardy due to its intrinsic difficulty. Early efforts for FSL were based on generative model that sought to build Bayesian probabilistic frameworks [24, 25]. As deep learning grew in popularity, more and more attentions were paid on meta-learning [26, 27]. We generally summarize current meta-learning methods for few-shot learning problems into five sub-categories: learn-to-measure, learn-to-finetune, learn-to-remember, learn-to-parameterize and learn-to-adjust.

The learn-to-measure methods attempt to learn to measure intra-class similarity and inter-class difference across different tasks, which include Matching Nets [9], ProtoNets [11], Mean Average Precision Networks (mAP Nets) [28], Relation

Nets [29], Task-dependent Adaptive Metric for Meta-learning (TADAM) [12] and Meta-learning with differentiable convex optimization (MetaOptNet) [30], etc. The learn-to-finetune methods suggest to fine-tune the base learner using the few support data and make the base learner converge fast on the few support data within several weight update steps, such as Meta-Learner LSTM [31], MAML [10], Latent Embedding Optimization (LEO) [32] and Meta Transfer Learning (MTL) [16], etc. The learn-to-remember methods include Memory-Augmented Neural Network (MANN) [33], Attentive recurrent comparators (ARCs) [34], Simple Neural Attentive Meta-Learner (SNAIL) [35] and Adaptive Posterior Learning (APL) [36], etc, and their primary idea is to model the support dataset as a sequence and formulate the FSL task as a sequence learning task, where the query data is required to match with the support data. The learn-to-parameterize methods learn to parameterize the module in the base learner to adapt to the novel tasks, and several typical methods include Model Regression Nets [37], Dynamic Nets [38], Acts2Params [39] and LGM-Net [40]. The learn-to-adjust methods advocate a task-specific adaptation by learning fast weights or learning neuron shifts, and several representative include Meta Networks [41], Metalearning with hebbian fast weights (MetaHebb) [42], Rapid Adaptation with Conditionally Shifted Neurons (CSN) [43], and Meta classifier-Predictor Module (MPM) [44], etc. The above methods, however, consistently need creating meta-training tasks on a large-scale supervised auxiliary dataset to obtain a FSL model.

2.2 Self-supervised/unsupervised methods for FSL

Compared to supervised FSL methods, the unsupervised FSL methods reduced the requirements for auxiliary data and they only leverage some unlabeled auxiliary data to forge the FSL model. Before our work, several methods, such as Clustering-based Pseudo-Labeling for Unsupervised Few-shot Learning (CACTUs) [45] and Unsupervised Meta-learning with Tasks Constructed by Random Labels (UMTRA) [46], provided insights to unsupervised solutions for FSL. CACTUs developed a two-stage strategy: constructing meta-training tasks on an unsupervised set by clustering algorithms and then running MAML [10] or ProtoNets [11] on the constructed tasks. Since the meta-training tasks are derived from unsupervised representations and the final FSL models are dominated by these tasks, CACTUs exhibit a strong dependence on unsupervised representation learning methods and clustering algorithms as well. Comparably, UMTRA proposed to construct meta-training tasks through augmenting the unsupervised data and treating the ancestor, on which augmentation is performed, and the corresponding augmented data as the same-concept data, which is followed by the ready-made MAML model. Both CACTUs and UMTRA, essentially, focused on how to allocate pseudo labels to unsupervised data such that the existing supervised FSL models can work without modification. Differently, our UTM does not hinge on any existing supervised FSL models, and instead it pursues a representation space from the unsupervised set wherein FSL can be conducted by a simple nearest neighbor classifier. It should be

noted that several self-supervised methods have been proposed recently, such as [47, 48], and they also used some unlabeled auxiliary data to facilitate the meta-train process, but they still rely on labeled auxiliary data and focus on semi-supervised FSL setting [21].

2.3 Unsupervised representation learning

Our UTM is based on unsupervised representation learning, a classical machine learning topic [49] which aims to acquire a pre-trained representation space from unsupervised data and works as a pre-bedding for downstream supervised learning tasks. In our work, several representative unsupervised representation learning methods developed recently, including BiGAN [50], ACAI [51] and DeepCluster [52], have been studied and compared with our UTM under the FSL testbed. Another unsupervised learning method similar in spirit to ours is Momentum Contrast (MoCo) [53], which contained a similar momentum update and entails complicated technical tricks to ensure its pretraining capability. Differently, we make a combination of momentum update with traction update and illustrate our advantage by a theoretical convergence analysis. It should be highlighted that we are the first to develop an unsupervised representation learning method customized for FSL problem.

3 Unsupervised tractive momentum

Instead of a labeled auxiliary set $\mathcal{A} = \{(x_i, y_i)\}_{i=1}^n$, we postulate that only an unlabeled $\mathcal{A} = \{x_i\}_{i=1}^n$ is attainable. The goal of UTM is to learn a representation space from the unlabeled $\mathcal{A}$ such that few-shot classification for novel concepts can be conducted via a simple nearest neighbor classifier. UTM contains two functional encoders, the traction encoder f_T parameterized by θ_T and the momentum encoder f_M parameterized by θ_M, who possess identical architecture but different parameters as well as different parameter updating rules. The traction encoder f_T and the momentum encoder f_M are both designed to extract representations from input images. We adopt the same architecture for both encoders to ensure consistent representational space and symmetric parameterization, which improves the stability of contrastive learning and facilitates encoder alignment. They map raw image x into normalized representation vector, that is, $||f_T(x|\theta_T)||_2 = 1$ and $||f_M(x|\theta_M)||_2 = 1$, $\forall x$. The output representations from both encoders are L2-normalized before computing similarity scores. This design ensures that the learned feature vectors are constrained on a hypersphere, enabling cosine similarity computation without scale distortion. Such normalization stabilizes contrastive gradients and aligns with common practice in self-supervised methods like SimCLR [54] and MoCo [53].

Algorithm 1 Unsupervised Tractive Momentum (UTM)

Require: unsupervised auxiliary set $\mathcal{A}=\{\ldots,x_i,\ldots\}$, number of keys per matching task N_K, traction step N_T, random augmentation function $\mathrm{Aug}(\cdot)$, traction rate α, momentum rate β, SGD learning rate lr.

randomly initialize θ_T, θ_M, metric scaling scalar μ
while not done **do**
  for $t = 1$ to N_T **do**
    sample N_K data $\{x_1, \ldots, x_{N_K}\}$ from $\mathcal{A}$
    randomly select x_j as postive data, $1 \le j \le N_K$
    augment x_j twice into $\mathrm{Aug}(x_j)$ and $\mathrm{Aug}'(x_j)$
    augment x_i into $\mathrm{Aug}(x_i)$, $\forall i \in \{1, \ldots, N_K\} \backslash j$
    query: $x^q \leftarrow \mathrm{Aug}'(x_j)$, *pos. key*: $x^+ \leftarrow \mathrm{Aug}(x_j)$ and *neg. keys*: $\{x_1^-, \ldots, x_{N_K-1}^-\} \leftarrow \{\mathrm{Aug}(x_i)\}_{i \neq j}$
    representation: $q=f_T(x^q|\theta_T)$, $k^+=f_M(x^+|\theta_M)$, and $k_i^-=f_M(x_i^-|\theta_M)$, $\forall i \in \{1, \ldots, N_K-1\}$
    evaluate task-specific metric loss $\mathcal{L}$ by Eq. (1)
    BP update: $(\theta_T, \mu) \leftarrow (\theta_T, \mu) - lr \cdot \nabla_{(\theta_T,\mu)}\mathcal{L}$
    momentum update: $\theta_M \leftarrow \beta \cdot \theta_M + (1-\beta) \cdot \theta_T$
  end for
  traction update: $\theta_T \leftarrow \alpha \cdot \theta_T + (1-\alpha) \cdot \theta_M$
end while

3.1 Self-supervised training on unsupervised data

The detailed algorithm of UTM is described in Algorithm 1. Given an unlabeled $\mathcal{A}$, UTM creates many synthetic query-key matching tasks on-the-fly by randomly sampling N_K data at a time and then augmenting them. A basic consideration is that two synthetic data, $\mathrm{Aug}(x)$ and $\mathrm{Aug}'(x)$, who are augmented from the same ancestor x, hold the same class label. In this case, one of the N_K data is randomly selected to be the positive data, and its two augmented data are cast as the query and the positive key, respectively, while the synthetic data augmented from the remainder $N_K - 1$ data are treated as negative keys. After that, the traction encoder f_T maps the query into q, and the momentum encoder f_M maps the positive key and the negative keys into k^+ and k_i^-, $i = 1, \ldots, N_K - 1$, respectively, followed by a metric loss:

$$\mathcal{L} = -\log \frac{\exp(\mu \cdot q^{\mathrm{T}} k^+)}{\exp(\mu \cdot q^{\mathrm{T}} k^+ + \sum_{i=1}^{N_K-1} \mu \cdot q^{\mathrm{T}} k_i^-)}, \tag{1}$$

where μ is a learnable metric scaling scalar [12] in the hope of facilitating metric training. The scalar μ functions as a temperature parameter to scale the cosine similarity, adjusting the sharpness of the distribution in Softmax, thereby modulating gradient magnitudes and influencing the contrastive learning dynamics. Combined with feature normalization, it helps prevent mode collapse and facilitates stable alignment between the two encoders, ultimately improving training robustness and performance. As a note, in our experiments, the metric loss $\mathcal{L}$ is evaluated based on multiple query data in a mini-batch manner.

3.1.1 Parameter updating mechanism

One straightforward idea is to make f_T and f_M parameter-sharing and synchronously update them based on metric loss, which collapses to an unsupervised version of ProtoNets [11]. However, it will easily render encoders sensitive to augmentation function rather than the underlying inter-concept divergence. We propose a combinatorial updating mechanism between θ_T and θ_M, as shown in line 11, 12 and 14 of Algorithm 1. For clarity, the parameter updating trajectory of two encoders is qualitatively depicted in Fig. 2. UTM makes f_T work like one engine that is driven by the metric loss $\mathcal{L}$ and intended to provide continuous traction to f_M, while keeps f_M updating in a slow momentum manner. To avoid excessive deviation between f_T and f_M after many matching tasks, UTM performs a traction update periodically to partly reset the f_T, which is updated violently, into the f_M, which moves forward steadily.

3.1.2 Way/shot-agnostic training

Most of supervised FSL methods, especially for meta-learning-based methods like MAML and ProtoNets, or the up-to-date unsupervised FSL methods like CACTUs and UMTRA, all need to customize their training configuration in light of each specific *N*-way *K*-shot setting to be tested. By contrast, our UTM is more elegant since its training is kept unaware of the specific form of test and it merely needs to be run once for each benchmark dataset.

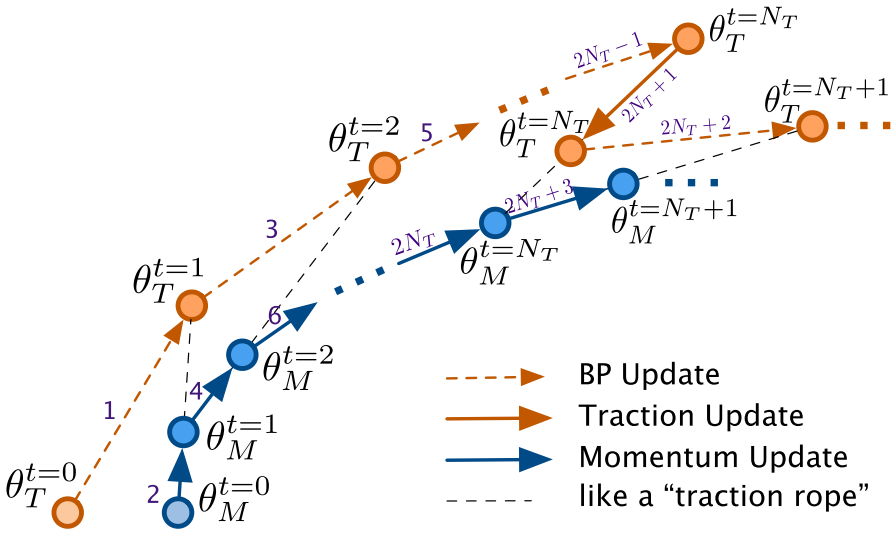

Fig. 2 Parameter updating trajectory (***brown***: traction encoder, ***blue***: momentum encoder). The $\theta^{t=i}$ denotes the parameters that are computed after i tasks. Numbers (***violet***) along with arrows show the order of parameter updates

3.2 Convergence analysis

Given the momentum update manner $\theta_M^t - \theta_T^t = \beta(\theta_M^{t-1} - \theta_T^t)$, it can be easily derived that, as $t \to \infty$, both θ_M^t and θ_T^t tend to converge together, otherwise the gap between them will continuously lead to parameter updating until convergence. We reorganize the momentum update rule into $\theta_M^t - \theta_T^t = \beta(\theta_M^{t-1} - \theta_T^{t-1}) + \beta\Delta\theta_T^t$ with $\Delta\theta_T^t = \theta_T^{t-1} - \theta_T^t$. Then,

$$\left|\left|\theta_M^t - \theta_T^t\right|\right|_2 \leq \beta\left|\left|\theta_M^{t-1} - \theta_T^{t-1}\right|\right|_2 + \beta\left|\left|\Delta\theta_T^t\right|\right|_2. \tag{2}$$

By Eq. (2), we can formalize the gap between f_T and f_M within N_T BP/momentum update steps between $[t_{n-1}, t_n^-]$ (see Fig. 3 for notation of update step index) as follows

$$\left|\left|\theta_M^{t_n^-} - \theta_T^{t_n^-}\right|\right|_2 \leq \beta^{N_T}\left|\left|\theta_M^{t_{n-1}} - \theta_T^{t_{n-1}}\right|\right|_2 + \eta\xi^{t_n}, \tag{3}$$

where $\eta = \frac{\beta(1-\beta^{N_T})}{1-\beta}$, $\xi^{t_n} = \max\left\{||\Delta\theta_T^t||_2, t \in (t_{n-1}, t_n^-]\right\}$. Next, we consider the whole update process between $[t_0, t_n]$ including traction update $\theta_T^{t_n} = \alpha\theta_T^{t_n^-} + (1-\alpha)\theta_M^{t_n^-}$, where $\theta_M^{t_n^-} = \theta_M^{t_n}$. The traction update can reduce the gap between two encoders momentarily since $\theta_M^{t_n} - \theta_T^{t_n} = \alpha(\theta_M^{t_n^-} - \theta_T^{t_n^-})$. Then, Eq. (3) can be further extended into

$$\begin{aligned}
&\left|\left|\theta_M^{t_n^-} - \theta_T^{t_n^-}\right|\right|_2 \leq \beta^{N_T}\left|\left|\theta_M^{t_{n-1}} - \theta_T^{t_{n-1}}\right|\right|_2 + \eta\xi^{t_n} \\
&= \alpha\beta^{N_T}\left|\left|\theta_M^{t_{n-1}^-} - \theta_T^{t_{n-1}^-}\right|\right|_2 + \eta\xi^{t_n} \leq \cdots \\
&\leq \alpha^{n-1}\beta^{nN_T}\left|\left|\theta_M^{t_0} - \theta_T^{t_0}\right|\right|_2 + \sum_{i=1}^{n}(\alpha\beta^{N_T})^{(n-i)}\eta\xi^{t_i}.
\end{aligned} \tag{4}$$

Obviously, $\lim_{n\to\infty}\alpha^{n-1}\beta^{nN_T}||\theta_M^{t_0} - \theta_T^{t_0}||_2 = 0$. For small i (in early training phase), its ξ^{t_i} is considerable since the training of f_T just starts, but $\lim_{n\to\infty, i\to 0}(\alpha\beta^{N_T})^{(n-i)} = 0$. For large i, $\lim_{n,i\to\infty}(\alpha\beta^{N_T})^{(n-i)} = 1$, but $\lim_{i\to\infty}\xi^{t_i} = 0$ since f_T is near saturation. Then, the second term of Eq. (4) satisfies $\lim_{n\to\infty}\sum_{i=1}^{n}(\alpha\beta^{N_T})^{(n-i)}\eta\xi^{t_i} = 0$. Consequently, we obtain $\lim_{n\to\infty}||\theta_M^{t_n} - \theta_T^{t_n}||_2 = 0$. It shows that the updating mechanism of UTM can guarantee a good convergence.

Comparably, if no traction update exists, we can get the following inequality by simply modifying step index in Eq. (3)

$$\left|\left|\theta_M^{t_n} - \theta_T^{t_n}\right|\right|_2 \leq \beta^n\left|\left|\theta_M^{t_0} - \theta_T^{t_0}\right|\right|_2 + \eta\xi, \tag{5}$$

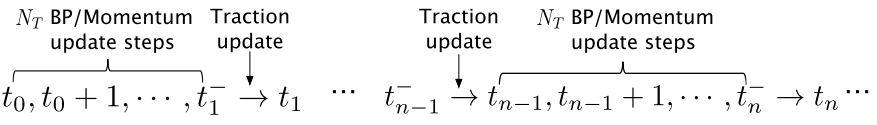

Fig. 3 Notations for parameter update step index

where $\eta = \frac{\beta(1-\beta^n)}{1-\beta}$, $\xi = \max\left\{||\Delta\theta_T^t||_2, t \in (t_0, t_n]\right\}$ (ξ is a considerable value that likely occurs during early training). Thus, $\lim_{n\to\infty} ||\theta_M^{t_n} - \theta_T^{t_n}||_2 \leq \frac{\beta}{1-\beta}\xi$, which underperforms the upper bound of convergence achieved by adding traction update.

We prove that the expected loss of UTM converges to a bound under mild conditions. Importantly, this convergence analysis provides insights into how UTM can maintain representational stability during parallel and asynchronous updates on multi-GPU architectures. The bounded divergence between encoder trajectories ensures that independent updates across GPUs remain within acceptable error margins, thus mitigating mode collapse in distributed learning scenarios.

3.3 Inference phase

Suppose the unsupervised training on the unlabeled $\mathcal{A}$ is completed, the stable momentum encoder f_M will be frozen to handle the N-way K-shot task in the nearest-neighbor form:

$$\begin{aligned} p_c &= \frac{1}{K}\sum_{i=1}^{NK} \mathbb{1}(y_i^s == c) f_M(x_i^s|\theta_M), \\ \hat{y}^q &= \arg\max_c \ S(f_M(x^q|\theta_M), p_c), \end{aligned} \tag{6}$$

where $c \in \{c_1, \ldots, c_N\}$ is class label, $\hat{y}^q$ is the predicted label, and $S(\cdot,\cdot)$ is a similarity metric (we adopt dot product). Although using f_T separately or the combination of f_T and f_M to make inference are two alternate choices, they leads to slightly attenuated performance (see ablation in Section 5).

4 Bound analysis

Assume $\mathcal{A}$ to be supervised, our framework can be trained in a supervised manner using this $\mathcal{A}$. We will show that the loss by UTM is an upper bound for the loss by supervised training, and prove that minimizing unsupervised loss makes sense.

Loss for Supervised Training. Let $\mathcal{C}$ denote the set of class label with prior distribution ρ. Assume that the augumented data $x \in \mathcal{X}$ is drawn from the data distribution $\mathcal{D}_c$, where $c \sim \rho$. Now considering one N-way task $\mathcal{T}_{sup}$ on N different classes $\mathcal{C}_{\text{sup}} = \{c_1, \ldots, c_N\}$, its multi-class classifier is denoted as the function $g : \mathcal{X} \to \mathbb{R}^N$. The softmax-based cross-entropy loss on data pair (x, y) can be rewritten as

$$\mathcal{L}_{\text{sup}}(g, x, y) = \log(1 + \sum_{y' \neq y} \exp(g(x)_{y'} - g(x)_y)), \tag{7}$$

where $g(x)_y$ is the y-th element of the vector $g(x)$. To qualify the two encoders f_T, f_M, we choose the classifier as $g(x)_c = \mu q^{\mathrm{T}} p_c$, where μ is a scaling scalar and $q = f_T(x|\theta_M)$ is the query representation, and $p_c = \mathbb{E}_{x\sim\mathcal{D}_c}[f_M(x|\theta_M)]$ is the mean of representation of inputs with label c. Then, the expected supervised loss in terms of f_T, f_M on N-way tasks is

$$\mathcal{L}_{\text{sup}} = \mathop{\mathbb{E}}_{c\sim\rho, x\sim\mathcal{D}_c} \log(1 + \sum_{c'\neq c} \exp(\mu q^{\mathrm{T}} p_{c'} - \mu q^{\mathrm{T}} p_c)). \tag{8}$$

Loss for UTM UTM assumes access to unsupervised task $\mathcal{T}$ with augumented data $\{x^q, x^+, x_1^-, \ldots, x_{N_K-1}^-\}$. We mark their ground-truth labels by $\mathcal{C}_{\text{U}} = \{c^q, c^+, c_1^-, \ldots, c_{N_K-1}^-\}$, respectively. Note that x^q and x^+ are drawn from the same data distribution $\mathcal{D}_{c^+}$ (since $c^q = c^+$) while negative x_i^- are from $\mathcal{D}_{c_i^-}$. Let $I = \{1, \ldots, N_K - 1\}$ be the set of indices of negative data, the unsupervised loss in Eq. (1) can be rewritten as

$$\mathcal{L}_{\text{U}} = \mathop{\mathbb{E}}_{q,k^+,k_i^-} \log(1 + \sum_{i\in I} \exp(\mu q^{\mathrm{T}} k_i^- - \mu q^{\mathrm{T}} k^+)). \tag{9}$$

It can be seen that Eq. (9) has a similar form with Eq. (8). As a first step, we show that $\mathcal{L}_{\text{U}}$ bounds the $\mathcal{L}_{\text{sup}}$ in an ideal situation. This conclusion might indicates that it makes sense to minimize the unsupervised UTM loss $\mathcal{L}_{\text{U}}$. We represent the upper bound for the supervised loss $\mathcal{L}_{\text{sup}}$ by:

Theorem 1 $\forall f_T, f_M \in \mathcal{F}$,

$$\mathcal{L}_{\text{sup}} \leq \gamma_0 \mathcal{L}_{\text{U}} + \delta, \tag{10}$$

where γ_0, δ are constants depending on the class distribution ρ. When ρ is uniform and $|\mathcal{C}| \to \infty$, then $\gamma_0 \to 1, \delta \to 0$.

Proof The key point for proof is the use of Jensen's inequality since $\ell(\boldsymbol{v}) = \log(1 + \sum_i \exp(\boldsymbol{v}_i)), \forall \boldsymbol{v} \in \mathbb{R}^{N_K-1}$ is a convex function ($\boldsymbol{v}_i$ is the i-th element of $\boldsymbol{v}$), that is,

$$\begin{aligned} \mathcal{L}_{\text{U}} &= \mathop{\mathbb{E}}_{q} \mathop{\mathbb{E}}_{k^+,k_i^-} \log(1 + \textstyle\sum_{i\in I} \exp(\mu q^{\mathrm{T}} k_i^- - \mu q^{\mathrm{T}} k^+)) \\ &\geq \mathop{\mathbb{E}}_{q,c^+,c_i^-} \log(1 + \textstyle\sum_{i\in I} \exp(\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+})). \end{aligned} \tag{11}$$

However, $\mathcal{C}_{\text{U}}$ may contains duplicate classes and even false negative classes. Clearly, we divide I into two disjoint subsets, true negative index set $I^- = \{i \in I | c_i^- \neq c^+\}$ and false negative index set $I^+ = \{i \in I | c_i^- = c^+\}$. We define $\mathcal{C}_{\text{uni}}$ as the label set after de-duplicating class labels in $\mathcal{C}_{\text{U}}$, $\mathcal{C}_{\text{uni}} \subseteq \mathcal{C}_{\text{U}}$. Since $\ell(\{\boldsymbol{v}_i\}_{i\in I_1\cup I_2}) := \log(1 + \sum_{i\in I_1\cup I_2} \exp(\boldsymbol{v}_i)) \geq \ell(\{\boldsymbol{v}_i\}_{i\in I_1}), \quad \forall \quad I_1, I_2 \subseteq I$, we could decompose Eq. (A4) into

$$\begin{aligned} &\mathop{\mathbb{E}}_{q,c^+,c_i^-} \log(1 + \textstyle\sum_{i\in I} \exp(\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+})) \\ &\quad \geq P(I^+ = \emptyset) \mathop{\mathbb{E}}_{q,c^+} \left[\ell\left(\{\mu q^{\mathrm{T}} p_c - \mu q^{\mathrm{T}} p_{c^+}\}_{c\in\mathcal{C}_{\text{uni}}\setminus c^+}\right) | I^+ = \emptyset\right] \\ &\qquad + P(I^+ \neq \emptyset) \mathop{\mathbb{E}}_{c^+} \left[\log(1 + |I^+|) | I^+ \neq \emptyset\right]. \end{aligned} \tag{12}$$

The first expectation in Eq. (A5) is actually the supervised loss $\mathcal{L}_{\text{sup}}$ in Eq. (8) by regarding $\mathcal{C}_{\text{sup}} := \mathcal{C}_{\text{uni}}$. Combining this result with Eq. (A4) (A5), we obtain the

inequality in Theorem 1 with $\gamma_0 = \frac{1}{P(I^+=\emptyset)}$, $\delta = -\frac{P(I^+\neq\emptyset)}{P(I^+=\emptyset)}\mathop{\mathbb{E}}_{c^+}\left[\log(1+|I^+|)|I^+\neq\emptyset\right]$. When ρ is uniform and $|\mathcal{C}|\to\infty$, then $P(I^+\neq\emptyset)\to 0$. □

Although we derive the above encouraging result about the relationship between UTM loss and supervised loss, however, for the task scenarios whose label space $|\mathcal{C}|$ is small (e.g., *mini*ImageNet), the $P(I^+\neq\emptyset)$ can never be close to zero and even the proportion of false negative data is considerable. In this case, minimizing $\mathcal{L}_{\mathrm{U}}$ will meet a theoretical bottleneck since $\mathcal{L}_{\mathrm{U}} \geq P(I^+\neq\emptyset)\mathbb{E}_{c^+}\left[\log(1+|I^+|)|I^+\neq\emptyset\right]$. Besides, Theorem 1 does not shed light into the explicit gap between $\mathcal{L}_{\mathrm{sup}}$ and $\mathcal{L}_{\mathrm{U}}$ as well as the underlying factor causing the gap. To overcome this issue, we further decompose the UTM loss $\mathcal{L}_{\mathrm{U}}$ into two terms: (1) $\mathcal{L}_{\mathrm{U}}^-$, the loss on all true negative data $x_i^-, i\in I^-$. (2) $\mathcal{L}_{\mathrm{U}}^+$, the loss on all false negative data $x_i^-, i\in I^+$. We define a notation of intra-class deviation as $s(f_M) := \mathbb{E}_c[\mathbb{E}_{x\in\mathcal{D}_c}\|f_M(x|\theta_M)-p_c\|_2^2]^{1/2}$, and show that $s(f_M)$ can bound $\mathcal{L}_{\mathrm{U}}^+$. Then, we get a new bound:

Theorem 2 $\forall f_T, f_M\in\mathcal{F}$,

$$\mathcal{L}_{\mathrm{sup}} \leq \gamma_0\mathcal{L}_{\mathrm{U}}^- + \gamma_1 s(f_M), \tag{13}$$

where γ_0, γ_1 are constants depending on the class distribution ρ. When ρ is uniform and $|\mathcal{C}|\to\infty$, then $\gamma_0\to 1, \gamma_1\to 0$.

Proof The key point for proof is that $\ell(\{\boldsymbol{v}_i\}_{i\in I_1\cup I_2}) \leq \ell(\{\boldsymbol{v}_i\}_{i\in I_1}) + \ell(\{\boldsymbol{v}_i\}_{i\in I_2}), \forall I_1, I_2\subseteq I$,

$$\begin{aligned}\mathcal{L}_{\mathrm{U}} &\leq \mathop{\mathbb{E}}_{q,k^+,k_i^-}\log(1+\textstyle\sum_{i\in I^-}\exp(\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+)) \\ &\quad + \mathop{\mathbb{E}}_{q,k^+,k_i^-}\log(1+\textstyle\sum_{i\in I^+}\exp(\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+)) \\ &:= \mathcal{L}_{\mathrm{U}}^- + \mathcal{L}_{\mathrm{U}}^+,\end{aligned} \tag{14}$$

where the first expectation is $\mathcal{L}_{\mathrm{U}}^-$ and the second one is $\mathcal{L}_{\mathrm{U}}^+$ (imagining that true negative data and false negative data have been separated during training). With the following inequalities $\ell(\{\boldsymbol{v}_i\}_{i\in I_1}) \leq \log(1+|I_1|) + \max\{\max\{\boldsymbol{v}_i\}_{i\in I_1}, 0\}$, and $\max\{\boldsymbol{v}_i\}_{i\in I_1} \leq |\max\{\boldsymbol{v}_i\}_{i\in I_1}| \leq \max\{|\boldsymbol{v}_i|\}_{i\in I_1} \leq \sum_{i\in I_1}|\boldsymbol{v}_i|$, we can get the following inequality:

$$\begin{aligned}\mathcal{L}_{\mathrm{U}}^+ &\leq \mathop{\mathbb{E}}_{q,k^+,k_i^-}\left[\log(1+|I^+|) + \textstyle\sum_{i\in I^+}(|\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+|)\right] \\ &= P(I^+\neq\emptyset)\mathop{\mathbb{E}}_{c^+}\left[\log(1+|I^+|)|I^+\neq\emptyset\right] \\ &\quad + \mathop{\mathbb{E}}_{q,k^+,k_i^-}\left[\textstyle\sum_{i\in I^+}|\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+|\right].\end{aligned} \tag{15}$$

By combining Eq. (10) with Eq. (A11) and Eq. (15), we get

$$\begin{aligned}\mathcal{L}_{\text{sup}} &\leq \gamma_0(\mathcal{L}_{\text{U}}^- + \mathcal{L}_{\text{U}}^+) + \delta \\ &\leq \gamma_0\mathcal{L}_{\text{U}}^- + \gamma_0 \mathop{\mathbb{E}}_{q,k^+,k_i^-}\left[\textstyle\sum_{i\in I^+} |\mu q^{\text{T}}k_i^- - \mu q^{\text{T}}k^+|\right] \\ &= \gamma_0\mathcal{L}_{\text{U}}^- + \gamma_0 \mathop{\mathbb{E}}_{c^+} \mathop{\mathbb{E}}_{x_q,x^+,x_i^-\sim\mathcal{D}_{c^+}}\left[\textstyle\sum_{i\in I^+} |\mu q^{\text{T}}k_i^- - \mu q^{\text{T}}k^+|\right].\end{aligned} \tag{16}$$

When the class distribution ρ is uniform, we have $\mathbb{E}|I^+| = (N_K - 1)/|\mathcal{C}|$ for any class. Considering that the representations are normalized to satisfy $\|q\|_2 = 1$, thus the right expectation in Eq. (16) can be bound by $s(f_M)$, that is,

$$\mathop{\mathbb{E}}_{c^+} \mathop{\mathbb{E}}_{x_q,x^+,x_i^-\sim\mathcal{D}_{c^+}}[\sum_{i\in I^+} |\mu q^{\text{T}}k_i^- - \mu q^{\text{T}}k^+|] \leq \sqrt{2}\mathbb{E}|I^+| \cdot s(f_M). \tag{17}$$

Combining Eq. (16) and Eq. (17), Theorem 2 can be proved, where $\gamma_1 = \frac{\sqrt{2}\gamma_0(N_K-1)}{|\mathcal{C}|}$, $\gamma_0 = \frac{1}{P(I^+=\emptyset)}$. When ρ is uniform and $|\mathcal{C}| \to \infty$, then $P(I^+ \neq \emptyset) \to 0, \gamma_0 \to 1, \gamma_1 \to 0$.

Compared to Theorem 1, Theorem 2 indicates that the supervised loss $\mathcal{L}_{\text{sup}}$ is bounded by two explicit parts. The first one is $\mathcal{L}_{\text{U}}^-$, which measures the similarity between the query data with the congener data as well as the difference between the query data with other distinct data. The second is $s(f_M)$, which acts as the penalty for representation ability by measuring the intra-class representation deviation. Moreover, the $\gamma_1 s(f_M)$ is an explicit gap distancing unsupervised UTM loss from supervised loss. Theoretically, if given an unsupervised set with infinite classes and data, the performance achieved by UTM can be very close to that by supervised training. □

5 Experiments

5.1 Datasets

We evaluate UTM on two FSL benchmark datasets, Omniglot [25] and *mini*ImageNet [9]. **Omniglot** is a character image dataset containing 1623 handwritten characters from 50 alphabets. Each character contains 20 gray images drawn by different writers. Following the standard few-shot learning protocol [25], we resize raw images into size of 28×28 and rotate each character by 0°, 90°, 180° and 270° to form 4 different classes. The 1200 characters with rotations (4800 classes) are for training, 100 characters (400 classes) for validation and 323 characters (1292 classes) for testing. ***mini***ImageNet is a downsampled image subset of the large-scale ImageNet [55]. It consists of 100 classes with 600 RGB images of size 84×84 per class, where 64 classes are for training, 16 classes for validation and 20 classes for testing. The above splits all keep the same with those used by CACTUs [45] and UMTRA [46] for comparison fairness. Certainly, the labels of data in training classes have been stripped to form the unsupervised auxiliary set $\mathcal{A}$. While *mini* ImageNet, a widely adopted benchmark in few-shot learning, is selected due to its standardized benchmarks and computational tractability, this does not reflect

limitations in our method. UTM is inherently scalable thanks to its dual-encoder design and supports parallel training on multi-GPU platforms, making it suitable for large-scale datasets.

5.2 Setup

For fairness, the architecture of encoder f_T, f_M keeps aligned with that used by CACTUs and UMTRA, as well as the supervised methods like MAML and ProtoNets. It is comprised of 4 convolutional blocks, each of which is a sequential combination of 64-channel 3×3 convolution, batch normalization, ReLU and 2×2 max-pooling. The last block is followed by a flattening and a normalization to form the feature representation, which leads to 64-/1600-dimensional representations for the images from Omniglot/*mini*ImageNet, respectively. We set $\alpha = 0.99$, $\beta = 0.999$ by monitoring validation performance. For augmentation Aug$(\cdot)$, we keep consistent with UMTRA: for Omniglot by randomly zeroing pixels and randomly shifting, while for *mini*ImageNet by the ready-made Auto-Augmentation [56] model. The SGD optimizer is used for BP update for f_T. All experiments are implemented in PyTorch and executed on an 8×A100 (40GB) GPU cluster using the NVIDIA Collective Communications Library for efficient multi-GPU communication. We adopt a distributed data parallel setup in which each GPU updates the traction encoder f_T independently, with periodic synchronization of the momentum encoder f_M. The UTM framework supports asynchronous computation, enabling scalable and stable contrastive pretraining across HPC resources.

5.3 Compared methods

Compared methods are explicitly divided into unsupervised group, supervised group and ablation group. Previous unsupervised FSL methods like CACTUs [45] and UMTRA [46], combined with representative unsupervised representation learning methods including BiGAN [50], ACAI [51] and DeepCluster [52], are compared with UTM. The supervised methods, MAML [10] and ProtoNets [11], are considered as the ceiling limit of the unsupervised methods. In ablation group, to explore the effectiveness of the parameter updating mechanism in UTM, four model variants in terms of parameter updating mechanism are studied: (1) Two encoders share the parameters and are updated synchronously (i.e., $\theta_T = \theta_M$). (2) Two encoders are updated separately via metric loss and no interaction exists between them. (3) f_T is fully reset into f_M after every N_T matching tasks (i.e., $\alpha = 0$). (4) f_T will never be reset towards f_M (i.e., $\alpha = 1$). In addition, we investigate two other ablation models that share the training phase with standard UTM but make inference by the single f_T or the combination of two encoders (f_T for query data, f_M for support data).

5.4 Results on Omniglot

Contrast results on Omniglot in Table 1 demonstrate that UTM completely surpasses CACTUs-MAML, CACTUs-ProtoNets, UMTRA and other alternate unsupervised

Table 1 Classification accuracy (%) on Omniglot (averaged over 1000 N-way K-shot (N,K) test tasks. •: best, ⋆: previous best)

Methods	**(5,1)**	**(5,5)**	**(20,1)**	**(20,5)**
Training from scratch	52.50	74.78	24.91	47.62
BiGAN k_{nn}-nearest neighbors	49.55	68.06	27.37	46.70
BiGAN linear classifier	48.28	68.72	27.80	45.82
BiGAN MLP with dropout	40.54	62.56	19.92	40.71
BiGAN cluster matching	43.96	58.62	21.54	31.06
BiGAN CACTUs-MAML	58.18	78.66	35.56	58.62
BiGAN CACTUs-ProtoNets	54.74	71.69	33.40	50.62
ACAI k_{nn}-nearest neighbors	57.46	81.16	39.73	66.38
ACAI linear classifier	61.08	81.82	43.20	66.33
ACAI MLP with dropout	51.95	77.20	30.65	58.62
ACAI cluster matching	54.94	71.09	32.19	45.93
ACAI CACTUs-MAML	68.84	87.78	48.09	76.36
ACAI CACTUs-ProtoNets	68.12	83.58	47.75	66.27
UMTRA	⋆83.80	⋆95.43	⋆74.25	⋆92.12
UTM ($\theta_T = \theta_M$)	86.13	94.30	67.83	83.67
UTM (no interaction)	84.72	93.46	66.01	81.62
UTM (fully reset θ_T)	91.73	97.95	79.74	93.70
UTM (never reset θ_T)	91.47	97.90	79.09	93.59
UTM (standard)	•92.00	•98.09	•79.99	•94.13
UTM (infer. by f_T)	90.71	97.76	77.58	93.32
UTM (infer. by $f_T + f_M$)	91.30	97.73	78.91	93.00
Supervised MAML	*94.46*	*98.83*	*84.60*	*96.29*
Supervised ProtoNets	*98.35*	*99.58*	*95.31*	*98.81*

methods, yielding dramatic improvements regardless of (N,K) settings: the best UTM model (i.e., standard UTM) outperforms previously state-of-the-art unsupervised performance by 8.20%, 2.66%, 5.74% and 2.01% for four settings, respectively. Another noticeable observation is the much smaller performance gap between UTM with supervised MAML and ProtoNets. For (5,5) setting, especially, UTM realizes accuracy 98.09% that is very close to accuracy 98.83% by supervised MAML, although it needs to use (4800×20+5×5) labeled data, whereas our UTM relies on only 5×5 labeled images for each (5,5) classification task.

5.5 Results on *mini*ImageNet

Table 2 contrasts UTM to other methods on *mini*ImageNet. Compared to Omniglot, the underlying complexity and ambiguity of the real-world image objects in *mini*ImageNet cause relatively lower classification accuracy. Nonetheless, for all the four settings, UTM still overmatches the previously best accuracy maintained by UMTRA or CACTUs-MAML, and obtains quite a competitive result to supervised FSL methods. These results provided a convincing proof for the effectiveness of the representation space learned by UTM.

Table 2 Classification accuracy (%) on *mini*ImageNet (averaged over 1000 N-way K-shot (N,K) test tasks. •: best, ⋆: previous best)

Methods	(5,1)	(5,5)	(5,20)	(5,50)
Training from scratch	27.59	38.48	51.53	59.63
BiGAN k_{nn}-nearest neighbors	25.56	31.10	37.31	43.60
BiGAN linear classifier	27.08	33.91	44.00	50.41
BiGAN MLP with dropout	22.91	29.06	40.06	48.36
BiGAN cluster matching	24.63	29.49	33.89	36.13
BiGAN CACTUs-MAML	36.24	51.28	61.33	66.91
BiGAN CACTUs-ProtoNets	36.62	50.16	59.56	63.27
DeepCluster k_{nn}-nearest neighbors	28.90	42.25	56.44	63.90
DeepCluster linear classifier	29.44	39.79	56.19	65.28
DeepCluster MLP with dropout	29.03	39.67	52.71	60.95
DeepCluster cluster matching	22.20	23.50	24.97	26.87
DeepCluster CACTUs-MAML	39.90	⋆53.97	⋆63.84	⋆69.64
DeepCluster CACTUs-ProtoNets	39.18	53.36	61.54	63.55
UMTRA	⋆39.93	50.73	61.11	67.15
UTM ($\theta_T = \theta_M$)	27.27	39.82	48.31	51.94
UTM (no interaction)	27.58	37.07	44.39	47.53
UTM (fully reset θ_T)	38.29	54.44	64.77	68.04
UTM (never reset θ_T)	38.27	54.92	64.73	68.04
UTM (standard)	•42.58	•59.13	•68.87	•71.92
UTM (infer. by f_T)	41.23	57.85	67.62	70.53
UTM (infer. by $f_T + f_M$)	41.59	58.63	67.78	69.74
Supervised MAML	*46.81*	*62.13*	*71.03*	*75.54*
Supervised ProtoNets	*46.56*	*62.29*	*70.05*	*72.04*

5.6 Analysis on discrepant advantage

We observe that the performance gap between UTM and supervised methods on Omniglot is relatively smaller than that on *mini*ImageNet. The reason can be partly blamed on the different $|\mathcal{C}|$ for two datasets. Concretely, the unsupervised $\mathcal{A}$ of *mini*ImageNet only contains 64 classes while that of Omniglot involves up to 4800 classes, which leads to a smaller γ_1 for Omniglot (larger $|\mathcal{C}|$ implies larger $P(I^+ = \emptyset)$, larger $P(I^+ = \emptyset)$ implies smaller γ_0, larger $|\mathcal{C}|$ + smaller γ_0 imply smaller γ_1). Assume that there is no significant difference about the representation ability of UTM on two datasets (even if the intra-class deviation on Omniglot is intuitively smaller than that on *mini*ImageNet since Omniglot is more easy), the gap $\gamma_1 s(f_M)$ on Omniglot is smaller than that on *mini*ImageNet. Certainly, the great disparity in data complexity between real-world *mini*ImageNet and gray-scale Omniglot is also one potential factor causing the discrepant advantage.

5.7 Ablation study

Ablation results are given in the penultimate part of Table 1 and 2. Especially, the first ablation model is equal to an unsupervised version of ProtoNets. It can be seen that the proposed parameter updating mechanism acquires the best accuracy among all alternate updating mechanisms. Another observation is that in most cases making inference by single f_M slightly outperforms that by single f_T or by the combination of f_T and f_M, which might benefit from the stable parameter updating and representation ability of f_M.

6 Conclusion and future works

In this work, we focus on weakening the prerequisite for creating a few-shot learner by distilling transferable knowledge in an unsupervised fashion. A novel unsupervised learning framework, Unsupervised Tractive Momentum (UTM) tailored for few-shot learning, is proposed. Convergence and bound analysis are given to prove the rationality of this novel learning paradigm from the theoretical perspective. Experimental results on two FSL benchmarks Omniglot and *mini*ImageNet show our method leads to significant improvement to state-of-the-art CACTUs and UMTRA, approaching supervised MAML and ProtoNets, albeit with only several labeled data. A limitation of this work is the ideal assumption that unsupervised auxiliary data and target FSL data come from the same data distribution domain. However, when the two come from different data domains, how to further optimize the UTM framework to tackle the unsupervised few-shot learning under cross-domain conditions will be a key direction for our future exploration.

7 Appendix Proof details

Proof of Theorem 1.

We first leverage the convexity of ℓ to get a lower bound of unsupervised loss $\mathcal{L}_{\mathrm{U}}$ in Eq. A4. Then we decompose the lower bound into a supervised loss $\mathcal{L}_{\mathrm{sup}}$ plus a degenerate term in Eq. A8.

Step 1. Convexity. $\ell(\boldsymbol{v}) = \log(1 + \sum_i e^{\boldsymbol{v}_i}), \forall \boldsymbol{v} \in \mathbb{R}^{N_K - 1}$ is a convex function. Because, $\forall t \in \mathbb{R}, \boldsymbol{z}, \boldsymbol{v} \in \mathbb{R}^{N_K - 1}$,

$$\begin{aligned} g(t) &= \ell(\boldsymbol{z} + t\boldsymbol{v}) = \log(1 + \textstyle\sum_i e^{\boldsymbol{z}_i + t\boldsymbol{v}_i}) \\ g'(t) &= \frac{\sum_i \boldsymbol{v}_i e^{\boldsymbol{z}_i + t\boldsymbol{v}_i}}{1 + \sum_i e^{\boldsymbol{z}_i + t\boldsymbol{v}_i}} \\ g''(t) &= \frac{(\sum_i \boldsymbol{v}_i^2 e^{\boldsymbol{z}_i + t\boldsymbol{v}_i})(1 + \sum_i e^{\boldsymbol{z}_i + t\boldsymbol{v}_i}) - (\sum_i \boldsymbol{v}_i e^{\boldsymbol{z}_i + t\boldsymbol{v}_i})^2}{(1 + \sum_i e^{\boldsymbol{z}_i + t\boldsymbol{v}_i})^2}, \end{aligned} \tag{A1}$$

we have,

$$\sum_i \boldsymbol{v}_i^2 e^{z_i+t\boldsymbol{v}_i} \geq 0, \tag{A2}$$

and Cauchy inequality,

$$\left(\sum_i \boldsymbol{v}_i^2 e^{z_i+t\boldsymbol{v}_i}\right)\left(\sum_i e^{z_i+t\boldsymbol{v}_i}\right) \geq \left(\sum_i \boldsymbol{v}_i e^{z_i+t\boldsymbol{v}_i}\right)^2. \tag{A3}$$

Thus, $g''(t)$ are always non-negative. $\ell(\boldsymbol{v})$ is a convex function.

Step 2. Jensen's inequality. The key point in the proof is the use of Jensen's inequality since $\ell(\boldsymbol{v}) = \log(1 + \sum_i \exp(\boldsymbol{v}_i)), \forall \boldsymbol{v} \in \mathbb{R}^{N_K - 1}$ is a convex function.

$$\begin{aligned}
\mathcal{L}_{\mathrm{U}} &= \mathop{\mathbb{E}}_{q,k^+,k_i^-} \log(1 + \sum_{i\in I} \exp(\mu q^{\mathrm{T}} k_i^- - \mu q^{\mathrm{T}} k^+)) \\
&= \mathop{\mathbb{E}}_{q,c^+,c_i^-} \mathop{\mathbb{E}}_{k^+\sim c^+, k_i^-\sim c_i^-} \log(1 + \sum_{i\in I} \exp(\mu q^{\mathrm{T}} k_i^- - \mu q^{\mathrm{T}} k^+)) \\
&\geq \mathop{\mathbb{E}}_{q,c^+,c_i^-} \log(1 + \sum_{i\in I} \exp(\mathop{\mathbb{E}}_{k^+\sim c^+, k_i^-\sim c_i^-} (\mu q^{\mathrm{T}} k_i^- - \mu q^{\mathrm{T}} k^+))) \\
&= \mathop{\mathbb{E}}_{q,c^+,c_i^-} \log(1 + \sum_{i\in I} \exp(\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+})).
\end{aligned} \tag{A4}$$

Note that in the upper bounded quantity, the random classes c_i^- may be false negative classes, that is, $c_i^- = c^+$.

Step 3. Decompose the lower bound. We divide all negative classes c_i^- set into two disjoint subsets, true negative classes and false negative classes. Clearly, we divide I into two unjoint subsets: $I^- = \{i \in I | c_i^- \neq c^+\}$ and $I^+ = \{i \in I | c_i^- = c^+\}$.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{q,c^+,c_i^-} \log(1 + \sum_{i\in I} \exp(\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+})) \\
&= \mathop{\mathbb{E}}_{q,c^+,c_i^-} \ell(\{\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+}\}_{i\in I}) \\
&= P(I^+ = \emptyset) \mathop{\mathbb{E}}_{q,c^+,c_i^-} \left[\ell(\{\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+}\}_{i\in I}) | I^+ = \emptyset\right] \\
&\quad + P(I^+ \neq \emptyset) \mathop{\mathbb{E}}_{q,c^+,c_i^-} \left[\ell(\{\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+}\}_{i\in I}) | I^+ \neq \emptyset\right],
\end{aligned} \tag{A5}$$

We define $\mathcal{C}_{\mathrm{uni}}$ as the label set after de-duplicating class labels in the tuple $\mathcal{C}_U$ (all labels including positive and negative classes). Since we have $\ell(\{\boldsymbol{v}_i\}_{i\in I_1\cup I_2}) := \log(1 + \sum_{i\in I_1\cup I_2} \exp(\boldsymbol{v}_i)) \geq \ell(\{\boldsymbol{v}_i\}_{i\in I_1}), \forall I_1, I_2 \subseteq I$, we can decompose the above quantity to handle repeated classes. If $I^+ = \emptyset$, then $I = I^-$, we can choose all de-duplicating negative class indexes as I_{uni}. Thus $I_{\mathrm{uni}} \subseteq I^- = I$, and $\ell(\{\boldsymbol{v}_i\}_{i\in I}) \geq \ell(\{\boldsymbol{v}_i\}_{i\in I_{\mathrm{uni}}})$. That is,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{q,c^+,c_i^-} \left[\ell(\{\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+}\}_{i\in I}) | I^+ = \emptyset\right] \\
&\geq \mathop{\mathbb{E}}_{q,c^+,c_i^-} \left[\ell(\{\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+}\}_{i\in I_{\mathrm{uni}}}) | I^+ = \emptyset\right] \\
&= \mathop{\mathbb{E}}_{q,c^+} \left[\ell(\{\mu q^{\mathrm{T}} p_c - \mu q^{\mathrm{T}} p_{c^+}\}_{c\in\mathcal{C}_{\mathrm{uni}}\setminus c^+}) | I^+ = \emptyset\right].
\end{aligned} \tag{A6}$$

Observe that the last expectation in Eq. A6 is actually the supervised loss $\mathcal{L}_{\text{sup}}$ by regarding $\mathcal{C}_{\text{sup}} := \mathcal{C}_{\text{uni}}$. The loss is based on the de-duplicating classes in our UTM.

If $I^+ \neq \emptyset$, all indexes in I^+ have the same labels as the positive class c^+, then $p_{c_i^-} = p_{c^+}$. Since $I^+ \subseteq I$, and $\ell(\{\boldsymbol{v}_i\}_{i\in I}) \geq \ell(\{\boldsymbol{v}_i\}_{i\in I^+})$. That is,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{q,c^+,c_i^-}\left[\ell(\{\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+}\}_{i\in I})|I^+ \neq \emptyset\right] \\
&\geq \mathop{\mathbb{E}}_{q,c^+,c_i^-}\left[\ell(\{\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+}\}_{i\in I^+})|I^+ \neq \emptyset\right] \\
&= \mathop{\mathbb{E}}_{q,c^+,c_i^-}\left[\ell(\{0\}_{i\in I^+})|I^+ \neq \emptyset\right] \\
&= \mathop{\mathbb{E}}_{q,c^+,c_i^-}\left[\log(1+|I^+|)|I^+ \neq \emptyset\right] \\
&= \mathop{\mathbb{E}}_{c^+}\left[\log(1+|I^+|)|I^+ \neq \emptyset\right].
\end{aligned} \tag{A7}$$

From Eq. A5, Eq. A6 and Eq. A7, we get

$$\begin{aligned}
&\mathop{\mathbb{E}}_{q,c^+,c_i^-} \log(1+\sum_{i\in I}\exp(\mu q^{\mathrm{T}} p_{c_i^-} - \mu q^{\mathrm{T}} p_{c^+})) \\
&\geq P(I^+ = \emptyset)\mathcal{L}_{\text{sup}} + P(I^+ \neq \emptyset)\mathop{\mathbb{E}}_{c^+}\left[\log(1+|I^+|)|I^+ \neq \emptyset\right].
\end{aligned} \tag{A8}$$

Combining Eq. A8 with Eq. A4, we have,

$$\mathcal{L}_{\text{U}} \geq P(I^+ = \emptyset)\mathcal{L}_{\text{sup}} + P(I^+ \neq \emptyset)\mathop{\mathbb{E}}_{c^+}\left[\log(1+|I^+|)|I^+ \neq \emptyset\right]. \tag{A9}$$

Thus we have proved the Theorem 1 using the fact that $\gamma_0 = \frac{1}{P(I^+=\emptyset)}$, $\delta = -\frac{P(I^+\neq\emptyset)}{P(I^+=\emptyset)}\mathop{\mathbb{E}}_{c^+}\left[\log(1+|I^+|)|I^+ \neq \emptyset\right]$.

Proof of Theorem 2.

First, we decompose the unsupervised loss into true negative samples loss $\mathcal{L}_{\text{U}}^-$ and false negative samples loss $\mathcal{L}_{\text{U}}^+$ by the property of Eq. A10, and further give an upper bound for the false negative samples loss $\mathcal{L}_{\text{U}}^+$ by the property of Eq. A13. Finally, we get a bound for our UTM in Eq. A17 and prove Theorem 2.

Step 1. Inequality 1 of ℓ. We note that the function ℓ satisfy the following constants: $\ell(\{\boldsymbol{v}_i\}_{i\in I_1\cup I_2}) \leq \ell(\{\boldsymbol{v}_i\}_{i\in I_1}) + \ell(\{\boldsymbol{v}_i\}_{i\in I_2}), \forall I_1, I_2 \subseteq I$. Because,

$$\begin{aligned}
\ell(\{\boldsymbol{v}_i\}_{i\in I_1\cup I_2}) &= \log(1+\sum_{i\in I_1\cup I_2} e^{\boldsymbol{v}_i}) \\
&\leq \log(1+\sum_{i\in I_1} e^{\boldsymbol{v}_i} + \sum_{i\in I_2} e^{\boldsymbol{v}_i}) \\
&\leq \log[(1+\sum_{i\in I_1} e^{\boldsymbol{v}_i})(1+\sum_{i\in I_2} e^{\boldsymbol{v}_i})] \\
&= \log(1+\sum_{i\in I_1} e^{\boldsymbol{v}_i}) + \log(1+\sum_{i\in I_2} e^{\boldsymbol{v}_i}) \\
&= \ell(\{\boldsymbol{v}_i\}_{i\in I_1}) + \ell(\{\boldsymbol{v}_i\}_{i\in I_2}).
\end{aligned} \tag{A10}$$

Step 2. Decompose the unsupervised loss. We have already divide all negative classes into two disjoint subsets and gotten their index sets I^-, I^+. I^- is for the true

negative samples while I^+ is for the false negative samples. According to these index sets and the property in Eq A10, we have,

$$\begin{aligned}
\mathcal{L}_{\mathrm{U}} &= \mathop{\mathbb{E}}_{q,k^+,k_i^-} \log(1+\sum_{i\in I}\exp(\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+)) \\
&= \mathop{\mathbb{E}}_{q,k^+,k_i^-} \ell(\{\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\}_{i\in I}) \\
&\le \mathop{\mathbb{E}}_{q,k^+,k_i^-} \big[\ell(\{\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\}_{i\in I^-}) \\
&\quad +\ell(\{\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\}_{i\in I^+})\big] \\
&= \mathop{\mathbb{E}}_{q,k^+,k_i^-} \big[\ell(\{\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\}_{i\in I^-})\big] \\
&\quad + \mathop{\mathbb{E}}_{q,k^+,k_i^-} \big[\ell(\{\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\}_{i\in I^+})\big] \\
&:= \mathcal{L}_{\mathrm{U}}^- + \mathcal{L}_{\mathrm{U}}^+,
\end{aligned} \tag{A11}$$

where the first expectation is $\mathcal{L}^-$ and the second is $\mathcal{L}^+$ (imagining that the true negative data and false negative data have been separated during training).

Step 3. Inequality 2 of ℓ. If the maximum value $v_{\max} := \max\{\boldsymbol{v}_i\}_{i\in I_1} > 0$, we have

$$\begin{aligned}
\ell(\{\boldsymbol{v}_i\}_{i\in I_1}) &= \log(1+\sum_{i\in I_1} e^{\boldsymbol{v}_i}) \\
&\le \log(1+|I_1|e^{v_{\max}}) \\
&= \log(1+|I_1|) + \log(e^{v_{\max}} + \tfrac{1-e^{v_{\max}}}{1+|I_1|}) \\
&\le \log(1+|I_1|) + v_{\max}.
\end{aligned} \tag{A12}$$

Otherwise, $\boldsymbol{v}_i \le v_{\max} \le 0, \forall i \in I_1$, we have $\ell(\{\boldsymbol{v}_i\}_{i\in I_1}) = \log(1+\sum_{i\in I_1} e^{\boldsymbol{v}_i}) \le \log(1+|I_1|)$. Thus we get the inequality,

$$\begin{aligned}
\ell(\{\boldsymbol{v}_i\}_{i\in I_1}) &\le \log(1+|I_1|) + \max\{v_{\max}, 0\} \\
&\le \log(1+|I_1|) + \sum_{i\in I_1}|\boldsymbol{v}_i|.
\end{aligned} \tag{A13}$$

Step 4. The upper bound of $\mathcal{L}_{\mathrm{U}}^+$. Using the property for the function ℓ in Eq. A13, we can get an upper bound for $\mathcal{L}_{\mathrm{U}}^+$,

$$\begin{aligned}
\mathcal{L}_{\mathrm{U}}^+ &= \mathop{\mathbb{E}}_{q,k^+,k_i^-} \big[\ell(\{\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\}_{i\in I^+})\big] \\
&\le \mathop{\mathbb{E}}_{q,k^+,k_i^-} \left[\log\left(1+|I^+|\right) + \sum_{i\in I^+}\left|\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\right|\right],
\end{aligned} \tag{A14}$$

where the second term acts as the penalty for representation ability by measuring the intra-class representation deviation.

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{q,k^+,k_i^-}\left[\sum_{i\in I^+}\left|\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\right|\right] \\
&= \mathop{\mathbb{E}}_{q,k^+,k_i^-}|I^+| \mathop{\mathbb{E}}_{q,k^+,k_i^-,i\in I^+}|\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+| \\
&\le |\mu| \mathop{\mathbb{E}}_{q,k^+,k_i^-}|I^+| \mathop{\mathbb{E}}_{q,k^+,k_i^-,i\in I^+}\sqrt{\|q\|_2^2\|k_i^- - k^+\|_2^2} \\
&\le |\mu| \mathop{\mathbb{E}}_{q,k^+,k_i^-}|I^+| \mathop{\mathbb{E}}_{q}\sqrt{\|q\|_2^2} \mathop{\mathbb{E}}_{k^+,k_i^-,i\in I^+}\sqrt{\|k_i^- - k^+\|_2^2} \\
&= |\mu| \mathop{\mathbb{E}}_{q,k^+,k_i^-}|I^+| \mathop{\mathbb{E}}_{q}\sqrt{\|q\|_2^2} \mathop{\mathbb{E}}_{k^+,k_i^-,i\in I^+}\sqrt{\|k_i^- - p_{c^+} + p_{c^+} - k^+\|_2^2} \\
&= \sqrt{2}|\mu| \mathop{\mathbb{E}}_{q,k^+,k_i^-}|I^+| \mathop{\mathbb{E}}_{q}\sqrt{\|q\|_2^2} \mathop{\mathbb{E}}_{k^+,k_i^-,i\in I^+}\sqrt{\|p_{c^+} - k^+\|_2^2}.
\end{aligned}
\tag{A15}
$$

All samples with indexes in I^+ have the same label c^+, then the expectation in Eq. A15 show that the intra-class representation deviation. Mark the deviation as $s(f_M) = |\mu| \mathop{\mathbb{E}}_{k^+,k_i^-,i\in I^+}\sqrt{\|p_{c^+} - k^+\|_2^2} = |\mu| \mathop{\mathbb{E}}_{c^+k^+}\sqrt{\|p_{c^+} - k^+\|_2^2}$. We have a uniform class distribution, then $\mathop{\mathbb{E}}_{q,k^+,k_i^-}|I^+| = (N_K - 1)/|\mathcal{C}|$. Considering that the representations are normalized to satisfy $\|q\|_2 = 1$, thus, the right expectation in Eq. A15 can be bound by $s(f_M)$, that is,

$$
\mathop{\mathbb{E}}_{q,k^+,k_i^-}\left[\sum_{i\in I^+}\left|\mu q^{\mathrm{T}}k_i^- - \mu q^{\mathrm{T}}k^+\right|\right] \le \sqrt{2} \mathop{\mathbb{E}}_{q,k^+,k_i^-}|I^+|s(f_M). \tag{A16}
$$

Combining Eq. A16, Eq. A14 and Eq. A11, we have

$$
\mathcal{L}_{\mathrm{U}} \le \mathcal{L}_{\mathrm{U}}^- + \sqrt{2} \mathop{\mathbb{E}}_{q,k^+,k_i^-}|I^+|s(f_M) + \mathop{\mathbb{E}}_{q,k^+,k_i^-}\left[\log\left(1 + |I^+|\right)\right], \tag{A17}
$$

and we have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{q,k^+,k_i^-}\left[\log\left(1 + |I^+|\right)\right] \\
&= P(I^+ \neq \emptyset) \mathop{\mathbb{E}}_{q,k^+,k_i^-}\left[\log\left(1 + |I^+|\right)\middle| I^+ \neq \emptyset\right] \\
&= P(I^+ \neq \emptyset) \mathop{\mathbb{E}}_{c^+}\left[\log\left(1 + |I^+|\right)\middle| I^+ \neq \emptyset\right]
\end{aligned}
\tag{A18}
$$

Combining Eq. A17, Eq. A18 and Theorem 1, we have proved Theorem 2 since $\gamma_1 = \sqrt{2}\gamma_0(N_K - 1)/|\mathcal{C}|$.

Acknowledgments This work was supported by National Natural Science Foundation of China (Grant No. 62506289) and Ministry of Science and Technology of the People's Republic of China (project number 2023ZD0506000).

Author contributions Zhong Cao: Conceptualization, theoretical derivation, experiments, writing and editing. Jiang Lu: Methodology, formalanalysis, validation, review, supervision. Liu He: Data curation, validation. Yuheng Luo: Investigation, distributed training.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The datasets Omniglot, *mini*ImageNet analyzed in our study are Benchmarked and public available in the public repositories: https://www.kaggle.com/datasets/qweenink/omniglot, https://www.kaggle.com/datasets/ctrnngtrung/miniimagenet/data.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436
2. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks, 3104–3112
3. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks, 1097–1105
4. Hinton G et al. (2012) Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Process. Magazine**29**
5. Carey S, Bartlett E (1978) Acquiring a single new word. Papers and Reports on Child Language Development 15:17–29
6. Biederman I (1987) Recognition-by-components: a theory of human image understanding. Psychol Rev 94:115–147
7. Clark EV First language acquisition (Cambridge University Press, 2009)
8. Koch G, Zemel R, Salakhutdinov R (2015) Siamese neural networks for one-shot image recognition, Vol. 2
9. Vinyals O, Blundell C, Lillicrap T, Wierstra D et al. (2016) Matching networks for one shot learning, 3630–3638
10. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks, 1126–1135
11. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning
12. Oreshkin B, López PR, Lacoste A (2018) Tadam: Task dependent adaptive metric for improved few-shot learning, 721–731
13. Bertinetto L, Henriques JF, Torr PH, Vedaldi A (2019) Meta-learning with differentiable closed-form solvers
14. Zhang R, Che T, Ghahramani Z, Bengio Y, Song Y (2018) Metagan: An adversarial approach to few-shot learning, 2365–2374
15. Chen Z, Fu Y, Chen K, Jiang Y-G (2019) Image block augmentation for one-shot learning 33:3379–3386
16. Sun Q, Liu Y, Chua T-S, Schiele B (2019) Meta-transfer learning for few-shot learning, 403–412
17. Wang Y, Yao Q, Kwok JT, Ni LM (2020) Generalizing from a few examples: A survey on few-shot learning. ACM Comput Surv 53:1–34
18. Lu J, Jin S, Liang J, Zhang C (2021) Robust few-shot learning for user-provided data. IEEE Trans Neural Net Learn Syst (TNNLS) 32:1433–1447
19. Ma Y et al (2021) Transductive relation-propagation with decoupling training for few-shot learning. IEEE Trans Neural Net Learn Syst (TNNLS) 33:6652–6664
20. Zhang Y et al. (2022) Graph information aggregation cross-domain few-shot learning for hyperspectral image classification. IEEE Trans. Neural Net. Learn. Syst.(TNNLS)

21. Lu J, Gong P, Ye J, Zhang J, Zhang C (2023) A survey on machine learning from few samples. Pattern Recogn 139:109480
22. Lu J, Xiao C, Zhang C (2024) Meta-modulation: A general learning framework for cross-task adaptation. IEEE Trans. Neural Net. Learn. Syst.(TNNLS)
23. Legg S, Hutter M (2007) Universal intelligence: A definition of machine intelligence. Mind Mach 17:391–444
24. Li F-F, Fergus R, Perona P (2006) One-shot learning of object categories. IEEE Trans Pattern Anal Mach Intell (TPAMI) 28:594–611
25. Lake BM, Salakhutdinov R, Tenenbaum JB (2015) Human-level concept learning through probabilistic program induction. Science 350:1332–1338
26. Bengio Y, Bengio S, Cloutier J (1990) Learning a synaptic learning rule. Université de Montréal, Département d'informatique et de recherche
27. Naik DK, Mammone R (1992) Meta-neural networks that learn by learning
28. Triantafillou E, Zemel R, Urtasun R (2017) Few-shot learning through an information retrieval lens
29. Yang FSY, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: Relation network for few-shot learning, 1199–1208
30. Lee K, Maji S, Ravichandran A, Soatto S (2019) Meta-learning with differentiable convex optimization, 10657–10665
31. Ravi S, Larochelle H (2017) Optimization as a model for few-shot learning
32. Rusu AA et al. (2019) Meta-learning with latent embedding optimization
33. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T (2016) Meta-learning with memory-augmented neural networks, 1842–1850
34. Shyam P, Gupta S, Dukkipati A (2017) Attentive recurrent comparators, 3173–3181
35. Mishra N, Rohaninejad M, Chen X, Abbeel P (2018) A simple neural attentive meta-learner
36. Ramalho T, Garnelo M (2019) Adaptive posterior learning: few-shot learning with a surprise-based memory module
37. Wang Y-X, Hebert M (2016) Learning to learn: Model regression networks for easy small sample learning, 616–634
38. Gidaris S, Komodakis N (2018) Dynamic few-shot visual learning without forgetting, 4367–4375
39. Qiao S, Liu C, Shen W, Yuille AL (2018) Few-shot image recognition by predicting parameters from activations, 7229–7238
40. Li H et al. (2019) Lgm-net: Learning to generate matching networks for few-shot learning, 3825–3834
41. Munkhdalai T, Yu H (2017) Meta networks, 2554–2563
42. Munkhdalai T, Trischler A (2018) Metalearning with hebbian fast weights. arXiv preprint arXiv:1807.05076
43. Munkhdalai T, Yuan X, Mehri S, Trischler A (2018) Rapid adaptation with conditionally shifted neurons, 3664–3673
44. Lai N, Kan M, Han C, Song X, Shan S (2020) Learning to learn adaptive classifier-predictor for few-shot learning. IEEE Trans Neural Net Learn Syst (TNNLS) 32:3458–3470
45. Hsu K, Levine S, Finn C (2019) Unsupervised learning via meta-learning
46. Khodadadeh S, Boloni L, Shah M (2019) Unsupervised meta-learning for few-shot image classification, 10132–10142
47. Gidaris S, Bursuc A, Komodakis N, Pérez P, Cord M (2019) Boosting few-shot visual learning with self-supervision, 8059–8068
48. Li X et al. (2019) Learning to self-train for semi-supervised few-shot classification. Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)
49. Barlow HB (1989) Unsupervised learning. Neural Comput 1:295–311
50. Donahue J, Krähenbühl P, Darrell T (2017) Adversarial feature learning
51. Berthelot D, Raffel C, Roy A, Goodfellow I (2018) Understanding and improving interpolation in autoencoders via an adversarial regularizer
52. Caron M, Bojanowski P, Joulin A, Douze M (2018) Deep clustering for unsupervised learning of visual features, 132–149
53. He K, Fan H, Wu Y, Xie S, Girshick R (2020) Momentum contrast for unsupervised visual representation learning, 9729–9738
54. Chen T, Kornblith S, Norouzi M, Hinton G A simple framework for contrastive learning of visual representations, 1597–1607 (PmLR, 2020)
55. Russakovsky O et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115:211–252

56. Cubuk ED, Zoph B, Mane D, Vasudevan V, Le QV (2018) Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501

Authors and Affiliations

Zhong Cao[1] · Jiang Lu[2] · Liu He[3] · Yuheng Luo[4]

✉ Zhong Cao
caoz10@foxmail.com

✉ Jiang Lu
lu-j13@tsinghua.org.cn

[1] Heidelberg Institute of Global Health in Heidelberg University, 69120 Heidelberg, Germany

[2] Advanced Technology and Equipment Research Institute, Xi'an Jiaotong University, Xi'an 710049, People's Republic of China

[3] School of Population Medicine and Public Health, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing 100730, People's Republic of China

[4] School of Health Policy and Management, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing, 100730, People's Republic of China