Appendix

Organization of the appendix. In this Appendix, we first prove the relationship between incorrect belief and conflicting belief in **Section A**. After that, we provide additional details of related works and comparison baselines in **Section B**. We then describe the training process with a complexity analysis in **Section C**. Finally, we provide additional results on label-efficient meta-learning, illustrative examples on the predicted multidimensional belief, effectiveness of the multidimensional belief showing its potential for detecting, OOD and uncertain predictions, comparison with other competitive baselines on any-shot and meta-dataset experiments, and ablation studies in **Section D**. Source code for our experiments is available at [1]

A. Proof Theorem 1

Before presenting the main proof, we first review some important concepts and show some useful results that will be used in the proof.

Definition 1. *Consider we have a sample for which the model outputs $N-$ dimensional belief vector $\mathbf{b} = (b_1, b_2, ...b_N)$. Let $S_b = \sum_{i=1}^{N} b_i$ represent the total belief, b_{cor} represent the correct belief, ib represent the incorrect belief, and cb represent the conflicting belief/dissonance. The conflicting belief cb can be computed from the belief vector as*

$$cb = \sum_{k=1}^{K} \Big(b_k \frac{\sum_{j \neq k} b_j Bal(b_j, b_k)}{\sum_{j \neq k} b_j} \Big), \tag{16}$$

$$Bal(b_j, b_k) = \begin{cases} 1 - \frac{|b_j - b_k|}{b_j + b_k}, & \text{if } b_i b_j > 0 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

where $Bal(\cdot, \cdot)$ is the relative mass balance function between two belief masses.

Proposition 1. *Zero belief masses in the belief vector have no contribution to both the conflicting belief and the incorrect belief. For any two non-zero belief masses b_i and b_j, the relative mass balance $Bal(b_i, b_j)$ is given by*

$$Bal(b_i, b_j) = \frac{2 \times \min(b_i, b_j)}{b_i + b_j} \tag{18}$$

Proposition 2. *The conflicting belief (cb) is a permutation invariant function over the belief vector ($\mathbf{b}$).*

Lemma 2. *The incorrect belief is an upper bound of $N-1$ belief subsets of the N-dimensional belief vector i.e. $ib \geq S_b - b_{max}$ where $b_{max} = max(b_1, b_2, ...b_N)$*

Proof. We know, $ib = S_b - b_{cor}$, and $b_{max} = \max(b_1, b_2, ...b_N) \geq b_{cor}$. $\implies ib \geq S_b - b_{max}$ □

Proof of Theorem 1.

Proof. Consider a sample in the task t. Due to Proposition 1, we can consider a belief vector with all non-zero beliefs for the proof. Further, as a consequence of Proposition 2, without loss of generality, we can consider that the beliefs in the belief vector are ordered in an descending order i.e. $\mathbf{b} = (b_1, b_2, ...b_N), b_{max} = b_1 \geq b_2, .. \geq b_N$ Now, the conflicting belief can be evaluated as

$$\begin{aligned} cb &= \sum_{i=1}^{N} b_i \times \Big(\frac{\sum_{j \neq i}^{N} b_j Bal(b_i, b_j)}{\sum_{j \neq i}^{N} b_j} \Big) \\ &= b_1 \Big(\frac{1}{\sum_{j \neq 1} b_j} \Big) \Big(\frac{2b_2^2}{b_1 + b2} + \frac{2b_3^2}{b_1 + b3} + ... \frac{2b_N^2}{b_1 + bN} \Big) \\ &+ b_2 \Big(\frac{1}{\sum_{j \neq 2} b_j} \Big) \Big(\frac{b_1 2b_2}{b_1 + b2} + \frac{2b_3^2}{b_2 + b3} + ... \frac{2b_N^2}{b_2 + bN} \Big) + ... \\ &+ b_N \Big(\frac{b_N}{\sum_{j \neq N} b_j} \Big) \Big(\frac{b_1 2b_N}{b_1 + b_N} + \frac{b_2 2b_N}{b_2 + B_N} + ... \frac{b_{N-1} 2b_N}{b_{N-1} + bN} \Big) \end{aligned} \tag{19}$$

[1] Link: https://github.com/pandeydeep9/Units-ML-CVPR-22

From the above expression, we can see that the numerator does not have a b_1^2 term. Considering the terms in dissonance with $2b_2^2$ in numerator, we get

$$
\begin{aligned}
b_2^2 \text{ terms} &= 2b_2^2\Big(\frac{b_1}{b_1+b_2}\Big)\Big(\frac{1}{\sum_{j\neq 1} b_j} + \frac{1}{\sum_{j\neq 2} b_j}\Big) \\
&= b2 \times 2b_2\Big(\frac{b_1}{b_1+b_2}\Big)\Big(\frac{b_1+b_2+2\times\sum_{j\neq 1,2} b_j}{\sum_{j\neq 1} b_j \sum_{j\neq 2} b_j}\Big) \\
&\quad \text{as } 2b_2 \leq b_1+b_2 \\
b_2^2 \text{ terms} &\leq b_2 b_1\Big(\frac{(2b_2+2\times\sum_{j\neq 1,2} b_j)}{b_1+b_2+2\times\sum_{j\neq 1,2} b_j}\Big) \times \Big(\frac{b_1+b_2+2\times\sum_{j\neq 1,2} b_j}{\sum_{j\neq 1} b_j \sum_{j\neq 2} b_j}\Big) \\
&\leq 2b_2 b_1 \frac{1}{\sum_{j\neq 2} b_j} \leq 2b_2
\end{aligned}
$$

Now, considering the terms in dissonance with $2b_n^2, n \in [3, N]$ in numerator, we get

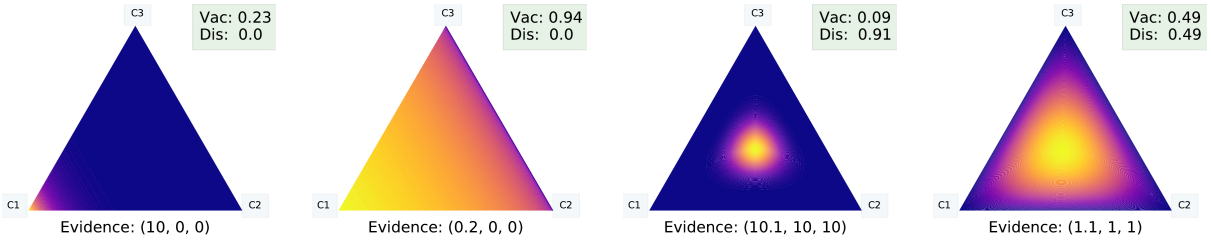
$$
\begin{aligned}
b_n^2 \text{ terms} &= 2b_n^2\Big(\frac{b_1}{b_1+b_n}\big(\frac{1}{\sum_{j\neq 1} b_j} + \frac{1}{\sum_{j\neq n} b_j}\big) + ... + \frac{b_{n-1}}{b_{n-1}+b_n}\big(\frac{1}{\sum_{j\neq n-1} b_j} + \frac{1}{\sum_{j\neq n} b_j}\big)\Big) \\
&\quad \text{as } 2b_n \leq b_1+b_n, ..., 2b_n \leq b_{n-1}+b_n \\
b_n^2 \text{ terms} &\leq 2b_n \frac{b_1}{\sum_{j\neq n} b_j} + 2b_n \frac{b_2}{\sum_{j\neq n} b_j} + ...2b_n \frac{b_{n-1}}{\sum_{j\neq n} b_j} \leq 2b_n
\end{aligned}
$$

We replace the bounds in the Equation (19) and use Lemma 2 to get

$$
cb \leq 2(b_2+b_3+...b_N) = 2(S_b - b_{max}) \leq 2ib \tag{20}
$$

which proves that for any sample, the incorrect belief is lower bounded by half the conflicting belief. Finally, task incorrect beliefs and the task conflicting beliefs are average of the query instance incorrect beliefs and conflicting beliefs respectably. The relationship in Eqn. (20) holds true for all query instances proving that the task incorrect belief is bounded by half of the conflicting belief on the task. □

B. Details of Related Work and Baselines



(a) A Confident Prediction (b) High vacuous belief (c) High conflicting belief (d) High Total Uncertainty

Figure 7. Subjective Logic Opinions: All 4 opinions correspond to a prediction of Class 1 with different uncertainty characteristics. Only the first opinion has low total uncertainty and can be trusted.

Subjective Logic Basics Subjective Logic (SL) is an extension of probabilistic logic [13], which considers the uncertainty in probability assignments along with the probabilities. Recently, using SL, deterministic deep learning (DL) models have been trained to output accurate confidence in predictions along with the predictions for both classification [31] [32] and regression [1] problems. For classification, the key idea is to train the DL models such that for any input, the model learns to output non-negative evidence for different classes. Using this evidence, the belief for different classes and the model's confidence can be calculated as:

$$
b_k = \frac{e_k}{\sum_{k=1}^{K} e_k + 1}, \quad v = \frac{K}{\sum_{k=1}^{K} e_k + 1}, \quad e_k \geq 0
$$

where e_k and b_k represent the evidence and belief for the k^{th} class and v represents the vacuity in the $K-$class classification problem. The vacuous belief (vacuity) is mainly due to the lack of evidence, is greatest when the model outputs no evidence, decreases as the model's evidence increases, and usually indicates unseen/out-of-distribution instances. Complimentary to the vacuous belief, conflicting belief (Eqn (6) *i.e.*, the dissonance) [14] indicates that the model is confused about the class assignment for the sample and is usually high for noisy/challenging data instances. Based on the conflicting and vacuous beliefs, we can decide which prediction to trust more. Moreover, for predictions with high uncertainty, we can infer the source of the uncertainty and take appropriate actions. Both vacuous belief and conflicting belief are instances of known uncertainty. Finally, there is unknown uncertainty (i.e. the uncertainty that the model is not aware of) that can be estimated from the incorrect beliefs. Unknown uncertainty is mainly due to highly confident incorrect predictions, can be quantified after obtaining the label information, and can only be estimated during training. The evidential models should be trained to minimize as much of the incorrect belief as possible.

We present an illustrative description of the subjective logic characteristics in Figure 7 for a 3 class classification (say between apples, mangoes, and oranges). If novel image (say a boat image) is shown to the model, the evidential model can express its lack of knowledge by outputting no evidence or equivalently high vacuity (Figure 7(b)). Similarly, for an image that is confusing among multiple classes, the model can output high evidence for all the classes leading to high dissonance (Figure 7(c)). Thus, accurately trained evidential models significantly boost the capabilities of deep learning models as they can identify the source of uncertainty, output level of confidence in predictions, and detect the data noise. Refer to [13] for a thorough study of SL and its characteristics.

Uncertainty-based Baselines. Meta-learning has recently been extended to the Bayesian setting [10, 12, 25]. Grant et al. [12] developed LLAMA, a bayesian extension of MAML, which used Laplace approximation to learn a distribution instead of the point estimate for task-specific parameters. LLAMA assumes the distribution for task-specific parameters to be gaussian which requires approximating the high-dimensional covariance matrix and thus may not scale to large networks. Finn et al., [7] presented Probabilistic MAML (PLATIPUS), which addresses the ambiguity of few-shot tasks by using principles of variational inference. Probabilistic MAML relies on a complex training procedure to learn a distribution only for the global parameters and resorts to point estimate for task-specific parameters. Gordan et al., presented VERSA, which uses an amortization network to output a distribution over weights of the base network, learns a distribution over task-specific parameters, and obtains the posterior predictive distribution [10]. Kim et al. presented Bayesian MAML, which employs an ensemble of MAML's to obtain uncertainty estimates [41]. Uncertainty awareness can be achieved by formulating meta-learning as a problem of inference in a hierarchical bayesian model as in ABML [28]. ABML uses amortized hierarchical variational inference for the task-specific distribution, learns a point-estimate for the prior distribution (global parameters), and uses Bayes by Backprop to obtain the task-specific distribution. In addition to the above uncertainty-aware meta-learning works, we consider the following baselines:

MAML [6] aims to learn the global parameters such that the model can adapt to new tasks using a few steps of gradient descent. MetaSGD [20] is an improvement over MAML where both the learning rate and learning direction are learned along with the good initialization. CAVIA [42] is another extension of MAML which addresses the meta-overfitting issue of MAML by separating the model parameters and updating only the subset of the parameters at test time. Reptile [26] is a first-order alternative to MAML trained for within task generalization. Finally, HSML [40] and MUMOMAML [38] are extensions of MAML designed to handle heterogeneous task settings. Moreover, our approach can be applied to most of the optimization-based approaches. We present an extension of our approach to MetaSGD in the additional experiments.

C. Training Process and Complexity Analysis

We aim to learn a good initialization such that for a new task, after learning from the support set, the meta-model can make accurate predictions as well as output the confidence in the prediction. MAML uses softmax activation at the final layer and cross-entropy loss that leads to the Maximum Likelihood Estimation of parameters. In Units-ML, for both local and global updates, we assume that the label for each sample in an N-way K-shot classification problem is obtained from a generative process with a Dirichlet prior: $\text{Dir}(\boldsymbol{p}_i|\boldsymbol{\alpha}_i)$ and a multinomial likelihood: $\text{Mult}(\boldsymbol{y}_i|\boldsymbol{p}_i)$.

In particular, for input $\boldsymbol{x}_i$ and $\mathbf{y}_i$ as the one-hot representation of the ground truth class, we treat outputs of the neural network as an evidence vector $\boldsymbol{e}_i = (e_i^1, e_i^2, ...e_i^N)^\top$. To ensure that the evidence is non-negative, we transform the final layer output by the Softplus function $\mathbf{e}_i = \ln(1 + e^{f(\mathbf{x}_i;\theta)})$. Further, we remove softmax function from our model as it squashes the model outputs in the range $[0, 1]$ which is too restrictive for the model evidence. With this, the parameters for the Dirichlet prior are calculated as $\boldsymbol{\alpha}_i = \boldsymbol{e}_i + \mathbf{1}$, following Eqn. (8). Similar to [31], we maximize the marginal likelihood obtained from

the Dirichlet prior and the multinomial likelihood.

$$\mathcal{L}_{mar}(\boldsymbol{x}_i, \boldsymbol{y}_i, \theta_m) = -\ln\Big(\int \text{Mult}(\boldsymbol{y}_i|\boldsymbol{p}_i)\text{Dir}(\boldsymbol{p}_i|\boldsymbol{\alpha}_i)d\boldsymbol{p}_i\Big) \;= -\sum_{j=1}^{N} y_i^j \ln(\frac{e_i^j+1}{\sum_{j=1}^{N} e_i^j + 1}) \tag{21}$$

Additionally, we want to train the model to output no incorrect belief that is achieved by using an incorrect belief regularization (Eqn. (11))

$$\mathcal{L}_{ib}(\boldsymbol{x}_i, \boldsymbol{y}_i) = \mathbf{b}_i \odot (\mathbf{1} - \mathbf{y}_i) \tag{22}$$

With this, the overall loss for one sample is given by:

$$\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i, \theta_m) = \mathcal{L}_{mar}(\boldsymbol{x}_i, \boldsymbol{y}_i) + \eta\mathcal{L}_{ib}(\boldsymbol{x}_i, \boldsymbol{y}_i) \tag{23}$$

Here, η is the regularization coefficient to balance between minimizing incorrect belief and minimizing the marginal likelihood under the Dirichlet prior, and θ_m represents the neural network parameters used to output evidence.

In our meta-learning setup, we consider a batch of tasks at each meta-iteration where each task has a support set and query set. We use the support set loss for local adaptation to task t in the inner loop using M steps of gradient descent as:

$$\begin{aligned}
&\theta_0^t = \theta \quad \text{[Make a copy of global parameters]} \\
&\theta_1^t = \theta_0^t - \alpha\nabla_{\theta_0^t}\mathcal{L}_t\big[f(\theta_0^t, X_S^t), Y_S^t\big] \qquad (24)\\
&\ldots \\
&\theta_M^t = \theta_{M-1}^t - \alpha\nabla_{\theta_{M-1}^t}\mathcal{L}_t\big[f(\theta_{M-1}^t, X_S^t), Y_S^t\big] \qquad (25)
\end{aligned}$$

At each gradient descent step, we define the support set loss as the average loss of $N \times K$ samples of the support set using the model parameters at that step

$$\mathcal{L}_t\big[f(\theta_{m-1}^t, X_S^t), Y_S^t\big] = \frac{1}{N \times K}\sum_{i=1}^{N\times K}\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i, \theta_{m-1}^t) \tag{26}$$

We perform this local adaptation for each of the I tasks at each meta-iteration. Next, we use the loss of the adapted model over query set samples to update the global parameters as

$$\theta \leftarrow \theta - \beta\sum_{t=1}^{I}\nabla_\theta\mathcal{L}_t\big[f(\theta_M^t, X_Q^t), Y_Q^t\big] \tag{27}$$

The query set loss is defined as the average loss of N_q query set instances with adapted model parameter

$$\mathcal{L}_t\big[f(\theta_M^t, X_Q^t), Y_Q^t\big] = \frac{1}{N_Q}\sum_{i=1}^{N_Q}\mathcal{L}(\boldsymbol{x}_i, \boldsymbol{y}_i, \theta_M^t) \tag{28}$$

Experiment and Dataset Details. We train the model using a batch of tasks at each iteration where the loss is given by Eqn. (28). We consider three datasets: Omniglot, CifarFS and Mini-ImageNet whose details are given in Table 2. We set $\eta = \min(8, 0.8 \times E)$ for all Omniglot experiments, $\eta = \min(2, 0.2 \times E)$ for CifarFS and 5-way 5-shot mini-ImageNet experiments, and $\eta = \min(0.5, 0.05 \times E)$ for 5-way 1-shot mini-ImageNet experiments where E is the epoch number. In the query set, unless specified, we consider 1 instance/class for omniglot and 2 instances/class for all other experiments. We consider a batch of 8 tasks at each iteration for Omniglot experiments, 4 tasks for CifarFS and 5-way 5-shot mini-ImageNet experiments and 2 tasks for 5-way 1-shot mini-ImageNet experiments. We consider 500 iterations/task and start task selection after 5 epochs. In active task selection, we consider a batch of 2 tasks at each iteration and increase E every 100 iterations, start task selection after 100 iterations, and set $\eta = \min(2, 0.2 \times E)$. Similar to Antoniou et al. [2] and ALFA [3], we learn the batch normalization parameters per step, learn the inner loop learning rate per layer and per step, use an ensemble of top 3 validation set models, and average the results from 3 independently run models to get the final test set performance in Table 1. These implementation tricks lead to some improvements for the Units-NTS model, with task selection further improving the generalization results. Furthermore, we start with balancing term λ value of $\lambda_{start} = 0.99$ and dynamically adjust it as $\lambda = \lambda_{start} - (\lambda_{start} - \lambda_{end}) \times \min(1.0, E/50)$ as training progresses to reach $\lambda_{end} = 0.5$ at the end of training. For local adaptation, we take 5 gradient descent steps in all Units experiments and 1 gradient descent step in all MetaSGD experiments.

Table 2. Dataset Properties

Characteristic	Omniglot	*mini*-ImageNet	CifarFS
Image Size	28×28	84×84	32×32
Channels	1	3	3
Total Classes	1623	100	100
Tr/Val Split	1150/50	64/16	64/16
Images/Class	20	600	600
Augmentation	Rotation	No	No

Algorithm and Complexity Analysis. We present the training process of Units-ML, the proposed task selection method in Algorithm 1. In Units-ML, we consider I multi-query tasks (see Fig. 1) with a total of J query sets such that minimal computation is required for determining the task uncertainty score. Our complexity analysis of MAML shows that the global update involves calculating Hessian gradient products and is computationally expensive, especially when we take many inner loop updates, or when the network has a large number of parameters. In particular, each additional inner loop update adds a $(I - H_k)$ term in the outer loop update. The Hessian calculation has an $O(D^2)$ complexity for a model with D parameters (where D is in the scale of thousands–millions in many typical deep neural networks). Even with the use of efficient Hessian-vector multiplication techniques [27], the computational cost would increase by $O(D)$ for one additional inner loop update. Furthermore, the computation is carried out at each meta-iteration for each task. With task selection, we ensure that tasks are informative for the global parameter update. For task selection, we only need predictions from the adapted model that adds little additional cost as compared to training on a new task. Specifically, task selection is independent of the number of inner-loop gradient descent steps in task adaptation and scales to any number of inner-loop updates without any additional computation. Tasks are selected using the informativeness score of the query set which requires one additional forward pass through the model. This introduces additional computation which scales linearly with the number of query sets considered in task selection. For a baseline model with the computational cost of $O(I \times M \times (F + B))$, the cost for the model with task selection is $O(I \times M \times (F + B) + J \times F)$ where both models adapt for M steps in the inner loop using I tasks (Task Selection model selects I query sets to be labeled from an unlabeled pool of J query sets), and the computational cost for forward pass through the model F is lower than the computational cost of the backward pass B.

Require: π, J: Initial training and sampled tasks ;
Require: $p(\mathcal{T})$: distribution over tasks ;
randomly initialize θ ;
while *not done* **do**
  if *Meta-Iteration* $\leq \pi$ **then**
    Sample I tasks $\mathcal{T}_i \sim p(\mathcal{T})$;
  else
    Sample I multi-query tasks with total of J unlabeled query sets $\mathcal{T}_i \sim p(\mathcal{T})$;
  end
  for ***each*** *support set* (X_S^i, Y_S^i)*:* **do**
    Compute task-specific parameters θ_M^i using M steps of gradient descent over the support set loss;
    Use the adapted model to select query set for each support set (X_S^i, Y_S^i) using Eqn. (13);
    Label the selected query sets
  end
  Update global parameter θ using query set loss of I selected query sets as
  $\theta \leftarrow \theta - \beta \sum_{i=1}^{I} \nabla_\theta L\big[f(\theta_M^i, X_Q^i), Y_Q^i\big]$;
end

Algorithm 1: Units-ML Task Selection for Efficient Meta-Training

D. Additional Experiments and Ablation Study

In this section, we first present additional results on label-efficient meta-learning. We then present illustrative examples of multidimensional belief quantification that demonstrate the usefulness of our model. Afterwards, we present our model's performance on any-shot datasets and a subset of Meta-Dataset. Finally, we present an ablation study to study the impact of incorrect belief regularization.

D.1. More Results on Label-Efficient Mata-Learning

To demonstrate the effectiveness of using multidimensional belief-based uncertainty measure for task selection, we consider a limited label meta-learning setting. We evaluate the models on a limited labeled budget scenario with a total of 10,000 tasks (each task has 1 instance/class in the query set for omniglot and 2 instances/class for all other datasets). We formulate the task as a multi-query task (with 8 query sets in each task). The baseline MAML model randomly selects the query set to be labeled. VERSA, an uncertainty-aware meta-learning model requests labels for the most informative query set based on the estimated query set uncertainty. Moreover, we extend MetaSGD to be evidential and uncertainty-aware using our proposed approach described in Section C (referred to as UA-MetaSGD). Both Units and UA-MetaSGD determine the task to be labeled based on the query set informativeness using Equation (13). All models are trained with a batch size of 2 for a total of 5000 iterations. Tables 3 and 4 show the results of the limited labeling budget experiments where the models with task selection show a clear advantage over the models without task selection especially when learning from a limited number of tasks.

Table 3. Meta-learning Performance Comparison under Limited Labeling budget Scenario - Omniglot

Omniglot 5w 1s	4000 Tasks	8000 Tasks	10,000 Tasks
MAML	73.44	80.18	85.56
Versa NTS	74.30	85.70	88.00
Versa TS	73.97	85.80	88.23
UA-MetaSGD NTS	85.06	88.64	89.76
UA-MetaSGD TS	87.08	90.50	91.62
Units-NTS	92.38	96.13	96.98
Units-ML	95.00	97.90	98.23
Omniglot 5w 5s	4000 Tasks	8000 Tasks	10,000 Tasks
MAML	92.00	96.26	96.38
Versa NTS	83.97	91.93	94.03
Versa TS	84.97	93.30	93.60
UA-MetaSGD NTS	93.28	94.96	95.38
UA-MetaSGD TS	94.32	97.00	97.48
Units-NTS	98.13	98.43	98.46
Units-ML	98.86	99.23	99.16
Omni 20w 1s	4000 Tasks	8000 Tasks	10,000 Tasks
MAML	65.24	72.41	73.91
Versa NTS	70.24	78.33	80.52
Versa TS	72.15	79.93	81.85
UA-MetaSGD NTS	61.94	71.34	71.44
UA-MetaSGD TS	62.38	71.07	72.51
Units-NTS	73.35	75.88	83.56
Units-ML	79.17	78.60	83.45
Omni 20w 5s	4000 Tasks	8000 Tasks	10,000 Tasks
MAML	86.62	89.75	88.26
Versa NTS	85.05	89.65	90.67
Versa TS	84.55	89.19	90.60
UA-MetaSGD NTS	73.65	78.82	79.94
UA-MetaSGD TS	75.09	79.52	81.74
Units-NTS	91.32	94.05	95.66
Units-ML	93.20	94.16	96.61

Table 4. Meta-learning Performance Comparison under Limited Label Budget - CifarFS and mini-ImageNet

CifarFS 5w 5s	4000 Tasks	8000 Tasks	10,000 Tasks
MAML	29.72	30.14	29.95
Versa NTS	32.60	40.11	42.91
Versa TS	33.38	42.06	43.75
UA MetaSGD NTS	52.18	54.38	58.30
UA MetaSGD TS	52.93	55.98	58.30
Units-NTS	53.87	58.37	61.30
Units-ML	55.88	60.53	61.39
***mini*-ImageNet** 5w 5s	4000 Tasks	8000 Tasks	10,000 Tasks
MAML	27.23	33.36	35.74
Versa NTS	39.45	45.16	45.48
Versa TS	37.86	45.48	46.21
MetaSGD NTS	44.88	48.95	51.69
MetaSGD TS	45.51	50.35	51.54
Units-NTS	43.21	48.32	49.54
Units-ML	47.33	48.34	54.23

D.2. Illustrative Examples of Predicted Multidimensional Belief

Mulit-Query Tasks. We present some qualitative results with a 5-way 2-shot *mini*-ImageNet Multi-Query tasks in Figure 8 to demonstrate the multidimensional belief characteristics for meta-learning. We assume that we have a limited labeling budget and can label only one of the two query sets. After learning on the support set, for query set 1 (Q1), the model can confidently predict the class labels with low overall task level uncertainties (both vacuous belief vb and conflicting belief cb). Q1 may not contribute much to the learning of the global parameters as the query set contains little new knowledge (indicated by low vacuous belief) and the model's class discriminating capabilities seem to be accurate (as indicated by low conflicting belief). If we consider query set 2 (Q2), then the model is highly uncertain about the predictions, with comparatively higher conflict in beliefs and a higher lack of confidence. Labeling Q2 to train the meta-learning model is likely to lead the meta-learning model to better generalization and label-efficient meta-learning.

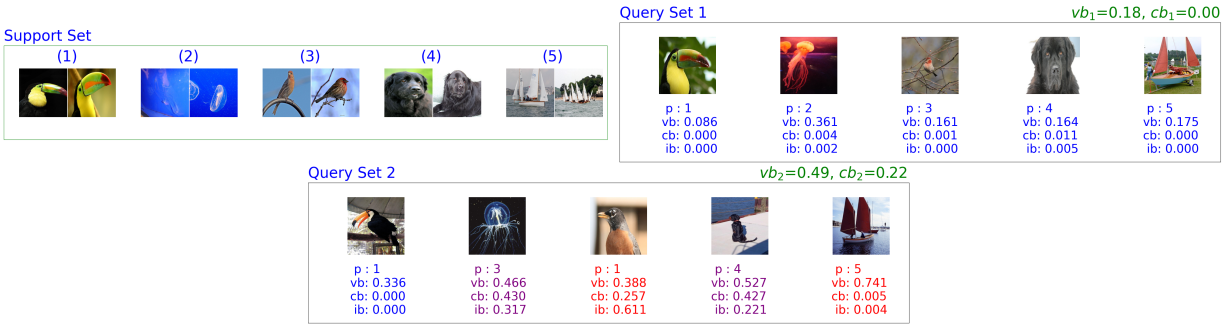


Figure 8. Multi-Query Task for Active Task Selection.

Uncertainty Prediction. We present some additional illustrative examples demonstrating our model's uncertainty prediction capabilities. We trained our model on 5-way 5-shot *mini*-ImageNet task and observed the model's behavior on 5-way tasks. Figure 9 shows the model's performance on a 5-way 1-shot task. Since each class in the support set has just 1 image/class, there might not be enough evidence in the support set to correctly predict all query set instances. This is reflected by the large model vacuity and dissonance for the query set instances. Further, due to limited evidence in the support set, the predictions for some query instances are wrong. For example, the wolf image in the query set is predicted as a lion. It may be because of the greater match of the orientation of the two animals. As we add instances in the support set that are helpful for the model to correctly classify the query set (see Figures 10, 11), the model starts to become be both confident in its prediction as well as correct its prediction indicated by a decrease in the query instance vacuity and dissonance. Further, the vacuity can be useful to detect open-set/OOD instances as shown in Figure 12.

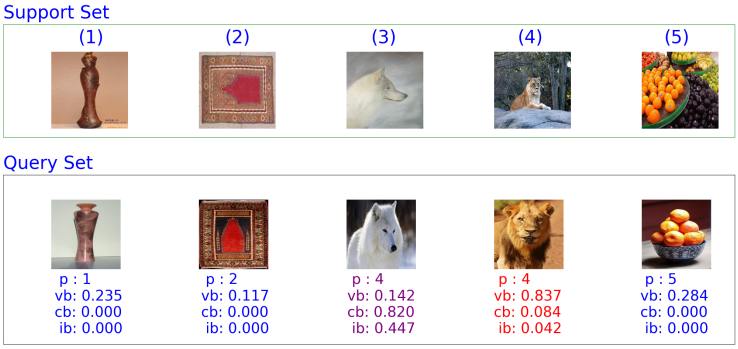


Figure 9. Uncertainty prediction in a 5-w 1-s task.

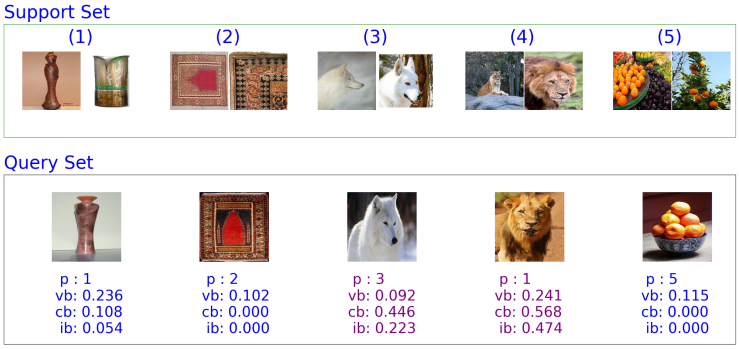


Figure 10. Uncertainty prediction in a 5-w 2-s task.

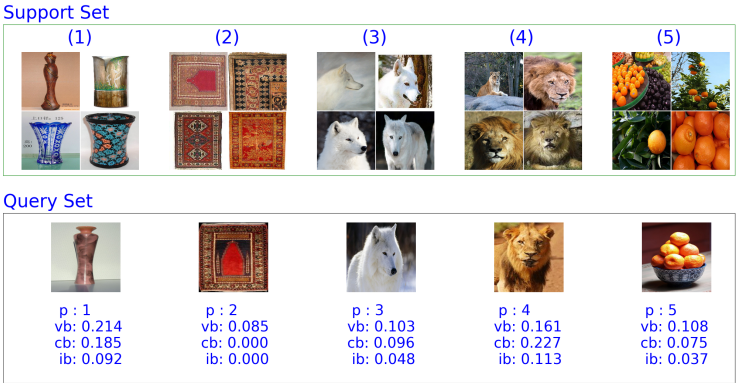


Figure 11. Uncertainty prediction in a 5-w 4-s task.

D.3. Effectiveness of Predicted Multidimensional Belief

Potential for OOD detection We perform experiments over 5-way 1-shot tasks on Omniglot to further demonstrate Units-ML's potential for Out-of-Distribution (OOD) detection. We train the model on a clean Omniglot dataset for 100 epochs and evaluate the model on 600 test tasks with query set samples rotated by various angles. Table 5 shows the model's performance on the query set after training for 100 epochs. Figure 13a shows the model's accuracy versus the predicted vacuity for different rotations of query set images. The accuracy drops with larger rotations on query images. Interestingly, vacuity increases proportionally which can be interpreted as: The model is aware of the shift in the distribution of the query set samples. We observe similar behavior with the scaling of the query set instances as shown in Figure 13b. Furthermore, in Figure 13a, due to the special nature of the Omniglot images (i.e., characters), some of the images (e.g., I,H,O,N,S,X,Z) are less sensitive to a 180° rotation. The model accurately recognizes this and reports a low vacuity around that angle.

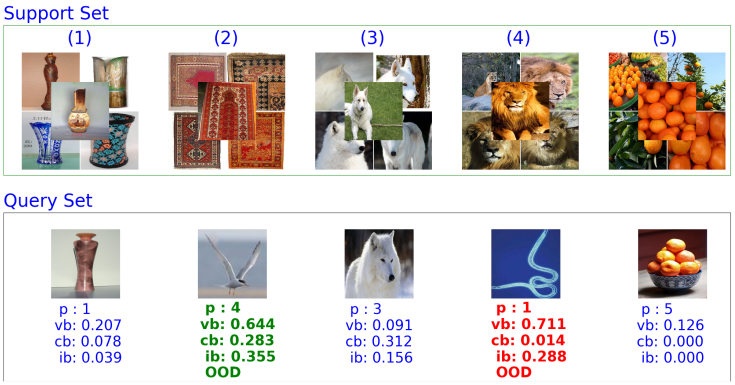
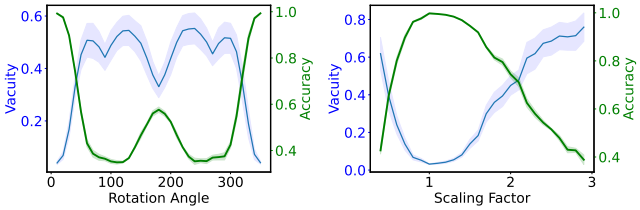


Figure 12. Uncertainty prediction in a 5-w 5-s *mini*-ImageNet test task with instances OOD instances in the query set.



(a) Rotated query set (b) Scaled query set

Figure 13. Vacuity and accuracy trends in OOD detection

Omniglot	5-way 1-shot(%)	5-way 5-shot(%)
MAML	98.7±0.4	99.1±0.1
Reptile	97.68±0.04	99.48±0.06
VERSA	99.70±0.20	99.75±0.13
Units-NTS	99.20±0.21	99.66±0.08
Units-ML	99.59±0.06	99.83±0.01

Table 5. Meta Learning Performance Comparison

Dataset wide statistics for OOD. We conduct additional experiments with 5-way 5-shot CifarFS-Aircraft and *mini*-ImageNet-CUB settings, where we consider the averaged performance across 600 test tasks. After training models on clean tasks from CifarFS and *mini*-ImageNet datasets, we compare the model's performance on the OOD query set constructed from Aircraft/CUB datasets with the In-distribution (InD) query set instances from CifarFS/*mini*-ImageNet, respectively. Specifically, during the test phase, we consider the support set from CifarFS/*mini*-ImageNet datasets and evaluate on the InD and OOD query sets. On average, the vacuity of InD query set is considerably lower than the vacuity of the OOD query set, which further justifies the potential of our model for OOD detection.

Dataset	Accuracy	InD Vacuity	OOD Vacuity
CifarFS	76.69%	0.18	0.45
mini-ImageNet	68.16%	0.29	0.48

Comparison with VERSA We also compare our Units-ML model with VERSA, another uncertainty aware model at different uncertainty thresholds for CifarFS dataset. We consider 5-way 5-shot CifarFS tasks where we use the output uncertainty from the two models to obtain top $T\%$ confident predictions over the query set of 600 test tasks and compare the performance. We use the variance of the predictions in VERSA to estimate the uncertainty. Units-ML achieves a higher prediction accuracy in all cases, which suggests that Unit-ML's predicted evidence-based uncertainty is more trustworthy. The VERSA model requires computationally expensive sampling to quantify uncertainty for each query set prediction. Moreover, using ideas from our work, the VERSA model can also be extended to be an computationally efficient evidential meta-learning model. We leave this extension as a future work.

Model	T=100%	T=70%	T=60%	T=50%
VERSA	74.69	78.33	81.41	84.33
Units-ML	76.50	83.26	86.36	87.23

D.4. Any-Shot and Multi-Dataset Experiments

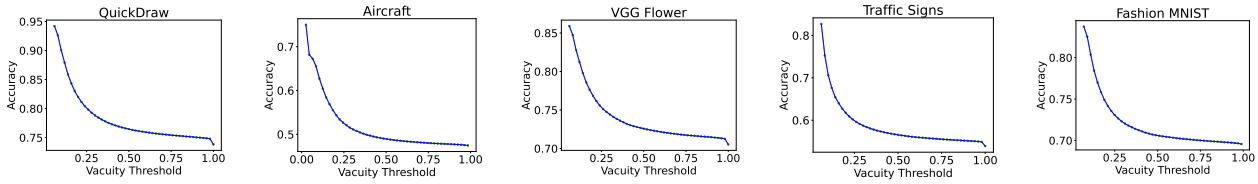
We also evaluated our Units model with any-shot classification tasks and multi-dataset settings using task/experiment setup as described in Lee et al. [19]. In any-shot experiments, we trained and evaluated on 5-way any-shot tasks with 15 instances/class in the query set having both class and task imbalance. A *mini*-ImageNet trained model was meta-tested on *mini*-ImageNet and CUB whereas a CIFARFS trained model was meta-tested on CIFARFS and SVHN test tasks. In multi-dataset experiments, the model was meta-trained using uniformly sampled 10-way any-shot tasks from Aircraft, QuickDraw, and VGG-Flower datasets and evaluated on Fashion-MNIST and Traffic Signs along with Aircraft, QuickDraw, and VGG-Flower datasets (tasks from Fashion-MNIST and Traffic Signs are not available to the model during meta-training phase). The results of the any-shot and multi-dataset experiments are presented in Table 6 and Table 7. For any-shot experiments, our model easily outperforms all the baselines except for Bayesian TAML [19]. In multi-dataset settings, the results are slightly different where our model outperforms all the baselines in three of the datasets: QuickDraw, Fashion-MNIST, and VGG-Flower. In the remaining two datasets, our model has comparable performance to other baselines and a slightly lower performance compared to Bayesian TAML. It is worth noting that Bayesian TAML is specifically designed to handle any-shot tasks with class and task imbalance but this is not the design goal of our model. Furthermore, as shown by Units-NTS 0.2/0.1 and Figure 14, if we consider the uncertainty threshold, our model can outperform all the baselines in all the settings.

Table 6. Any-Shot Setting Comparison

Meta-Training	*mini*-ImageNet		CifarFS	
Meta-Testing	*mini*-ImageNet	CUB	CifarFS	SVHN
MAML	66.64	65.77	71.55	45.17
Meta-SGD	69.95	65.94	72.71	46.45
fo-Proto-MAML	68.96	61.77	71.80	40.16
Bayesian TAML	71.46	71.71	75.15	51.87
Units-NTS	71.70	67.95	76.76	52.11
Units-NTS 0.2	83.27	80.01	84.60	70.63
Units-NTS 0.1	92.18	88.82	92.00	81.47

Table 7. Multi-Dataset Setting Comparison

Meta-Training	Aircraft, QuickDraw, and VGG-Flower				
Meta-Testing	Aircraft	QuickDraw	VGG-FLower	Traffic Signs	Fashion-MNIST
MAML	48.60	69.02	60.38	51.96	63.10
Meta-SGD	49.71	70.26	59.41	52.07	62.71
fo-Proto-MAML	51.15	69.84	65.24	53.93	63.72
Bayesian TAML	54.43	72.03	67.72	64.81	68.94
Units-NTS	46.88	73.82	70.52	53.90	69.09
Units-NTS 0.2	54.64	82.00	76.84	61.74	74.49
Units-NTS 0.1	63.43	90.06	82.78	70.60	81.49

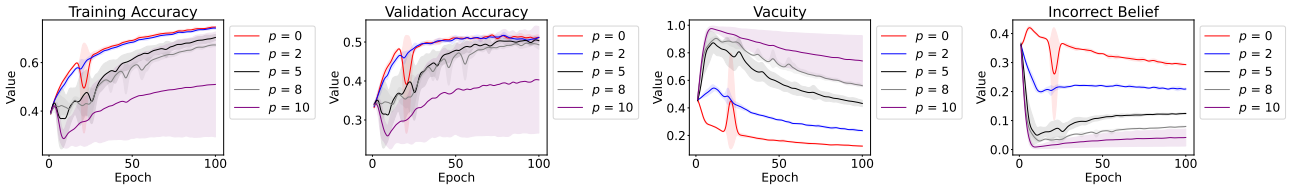


(a) QuickDraw (b) Aircraft (c) VGG Flower (d) Traffic Signs (e) Fashion MNIST

Figure 14. Impact of Vacuity Threshold for Different Datasets in Meta-Dataset

D.5. Effect of Incorrect Belief Regularization

We add a belief regularization term (Eqn. (23)) to encourage our model to output low (ideally no) belief for classes other than the ground truth and ensure low incorrect belief. The effect of the regularization term is controlled by $\eta = \min(p, p * E/10)$ (Eqn. (23)) where we p is a hyperparameter. Here, we study the impact of this hyperparameter on our model's accuracy, vacuous belief, and incorrect belief. Figure 15 shows the impact of regularization on training accuracy, validation accuracy, vacuity, and incorrect belief for a 5-way 1-shot CifarFS experiment. If there is low/no regularization, then the model outputs high confidence even for wrong predictions as indicated by large incorrect beliefs. When the incorrect belief regularization dominates the loss, the model outputs high vacuity for all tasks and the model fails to train properly. The model shows the best performance when there is a good balance in penalizing incorrect belief (through belief regularization) and encouraging large correct belief (through the loss term).



(a) Training (b) Validation (c) Vacuity (d) Incorrect Belief

Figure 15. Impact of regularization on (a) Training Accuracy, (b) Validation Accuracy, (c) Training Vacuity, and (d)Training Incorrect Belief for 5-way 1-shot CifarFS experiment

Sensitivity to λ. The model should focus on acquiring new knowledge (most vacuous tasks) at the initial phase, and as training progresses, transition to correct its acquired but incorrect knowledge. Thus, we set λ heuristically to take a relatively large value and gradually decrease as training progresses. Specifically, in all Units-ML experiments, we start with balancing term λ value of $\lambda_{start} = 0.99$ and dynamically adjust it as $\lambda = \lambda_{start} - (\lambda_{start} - \lambda_{end}) \times \min(1.0, E/50)$ as training progresses to reach $\lambda_{end} = 0.5$ at the end of training. Since the vacuous belief (vb^t) also decreases as the model explores the task space, the performance is quite robust as shown in Figure 16, where we tested different λ values on 5-way 1-shot tasks using the Omniglot dataset.

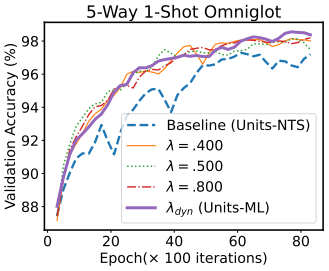


Figure 16. Impact of λ