META-LEARNING WITH EXPLICIT TASK INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A common approach in few-shot learning is to adapt to a new task after learning a variety of similar tasks. When the diversity of the tasks is high, however, it can be challenging for models to generalize effectively. Prior work has approached this problem by inferring task information implicitly from the data in order to better adapt to each new task. However, in some cases, explicit information about tasks is available that can inform task adaptation to improve performance, especially in the context of few-shot learning. In this work, we introduce task-informed meta-learning (TIML), an algorithm which modulates a model based on explicit task metadata. We evaluated TIML for a range of classification and regression tasks and found that TIML significantly improves performance in both regimes across a diversity of model architectures. In particular, we show the power of TIML in remote sensing for agriculture—an area of high societal impact where traditional methods have failed due to limited and imbalanced data.

1 INTRODUCTION

When learning from limited data, a common approach is to learn from similar tasks and adapt to each new task (Thrun & Pratt, 1998; Wang et al., 2020). Model-Agnostic Meta-Learning (Finn et al., 2017) is a popular method for training a neural network on a set of tasks that promotes rapid generalization to other tasks. This approach has seen success in a variety of domains with plentiful subsequent studies (Gui et al., 2018; Antoniou et al., 2019; Rajeswaran et al., 2019).

When tasks are highly heterogeneous and imbalanced, however, meta-learning models struggle to learn to generalize to new tasks (Triantafillou et al., 2020). In particular, the model can experience degraded performance on tasks that are out-of-distribution relative to the training tasks seen by the model (Collins et al., 2020). This has negative consequences for the downstream use of models, for example leading to problems of model inequity since many datasets are focused on limited geographies, such as North America and Europe (Shankar et al., 2017; Koch et al., 2021). Prior work has attempted to address such problems by inferring information about a task from its samples implicitly and using it to adapt the model to the task (Vuorio et al., 2019; Triantafillou et al., 2021), or by minimizing the difference in performance across tasks (Jamal & Qi, 2019).

In many settings, however, explicit metadata about tasks that could provide useful context for task adaptation is readily available. For example, remote sensing images typically have location metadata (latitude and longitude); inferring such metadata from the images themselves is difficult and unnecessary. Providing such metadata explicitly to the model could enable the model to learn more efficiently than if it was forced to attempt to infer task-level information from examples. An approach that provides explicit task metadata could also improve performance on rare tasks by allowing the model to better learn the task distribution. This line of inquiry aligns with prior work that has shown the value of integrating metadata into machine learning algorithms in other contexts (You et al., 2017; Xie et al., 2020; Ayush et al., 2021; Mac Aodha et al., 2019).

We present *Task-Informed Meta-Learning* (TIML), a task-adaptive meta-learning method which modulates the meta-learner using explicit task metadata. TIML learns embeddings from task metadata that inform the internal representation space of the meta-learner prior to task-specific fine-tuning. We demonstrate the efficacy of TIML on classification and regression datasets across a range of models and domains (Lake et al., 2015; Tseng et al., 2021b; You et al., 2017), showing

a significant improvement over traditional task-adaptive (Vuorio et al., 2019) and metadata-aware methods (You et al., 2017).

In particular, we show the power of TIML in tasks involving geospatial and remote sensing datasets, a domain in which i) metadata is prevalent (Xie et al., 2020), ii) limited labelled data is an acute challenge (Wang et al., 2018; Kerner et al., 2020; Jean et al., 2016), and iii) data (and therefore tasks) are highly spatially imbalanced (de Vries et al., 2019; Shankar et al., 2017). We conducted experiments for crop classification and agricultural yield estimation, which are challenging and highly impactful tasks that provide vital information for combating food insecurity globally, especially as climate change threatens crop production. We used datasets created specifically to inform agricultural policy (Tseng et al., 2021b) for which traditional methods have significantly underperformed, since data are sparse and imbalanced across crop types and geographies. TIML significantly outperforms all other approaches, showing excellent performance even for extremely small training datasets. We also show the superior performance of TIML on the Omniglot dataset (Lake et al., 2015) to demonstrate its broad applicability across domains.

2 Related Work

Meta-Learning Meta-learning, or *learning to learn*, consists of learning to solve a task after having seen other example tasks (Thrun & Pratt, 1998). Recent work in this area has focused on few-shot learning, i.e., learning the new task with few training datapoints. Ravi & Larochelle (2017) learn an optimizer from a set of tasks which can then be applied in a new task, while Snell et al. (2017) cluster data samples in the embedding space to perform few-shot classification. We leverage model-agnostic meta-learning (MAML) (Finn et al., 2017), a few-shot meta-learning framework that uses example tasks to learn a set of initial weights that can rapidly generalize to a new task.

Task-adaptive meta-learning Task-adaptive meta-learning aims to tailor meta-models to the specific tasks they will be fine-tuned on. In particular, numerous methods adapt the model weights learned during gradient-based meta-learning using task information inferred from the available training task samples by modulating the model parameters (Lee et al., 2019; Vuorio et al., 2019; Triantafillou et al., 2021; Yao et al., 2019; Oreshkin et al., 2018; Rusu et al., 2018), varying the task learning rates (Lee et al., 2019), or adapting the optimizer (Simon et al., 2020; Baik et al., 2020a) or loss function (Baik et al., 2021). A critical component of these methods is some representation of task i, which we denote as t_i, to inform the task adaptation. Previous approaches have attempted to infer t_i from the available training samples in a task. However, such approaches typically require many datapoints to infer task similarity and also assume that tasks are readily identifiable from training samples, which is not always true. We investigate the utility of *explicit* task metadata to represent t_i, instead of *implicitly* inferring t_i from the data. Unlike approaches that learn task information implicitly, our approach enables few-shot or even zero-shot learning, in which no labelled training data is available (Appendix B).

Learning from metadata There have been several recent studies on how metadata—or auxiliary data—can inform machine learning algorithms. A common approach is to use an additional data source to learn initial weights that are useful for a range of downstream tasks (Ayush et al., 2021; Xie et al., 2020; Jean et al., 2019). Alternatively, metadata may be integrated into the learning process, typically by updating the final predictions of the model to align them with information provided by the metadata (You et al., 2017; Mac Aodha et al., 2019; Kluger et al., 2021). We investigate the usefulness of metadata in few-shot learning and in particular to inform gradient-based meta-learning.

Few-shot learning in geospatial contexts Few-shot learning has been extensively explored in geospatial machine learning, particularly by attempting to transfer knowledge from data-rich to data-sparse regions. This has been achieved in a variety of ways, ranging from transfer learning (Wang et al., 2018; Jean et al., 2016) to multi-task learning (Kerner et al., 2020; Chang et al., 2019) to meta-learning (including MAML) (Rußwurm et al., 2020; Tseng et al., 2021a;b). We consider how models may be adapted to an unseen target task, which can prevent performance degradation for unseen tasks observed by prior studies in this domain (Rußwurm et al., 2020). In addition, there is plentiful metadata available in geospatial machine learning tasks (Xie et al., 2020); we investigate how such metadata can be used to improve performance.

3 TASK-INFORMED META-LEARNING

Model-Agnostic Meta-Learning (MAML) learns model weights θ that are close to optimal for each of a variety of different tasks, allowing the optimal weights for a specific task to be reached with little data and/or few gradient steps. These weights θ are updated by fine-tuning them on a training task (inner loop training), yielding updated weights θ'. A gradient for θ is then computed with respect to the loss of the updated model, $L_{\theta'}$ which is used to update θ (outer loop training).

Our approach, Task-Informed Meta-Learning (TIML) (Algorithm 1), builds on MAML by leveraging explicit task-level metadata. We introduce a task encoder to modulate the weights of the meta-learner. We also introduce *forgetfulness*, a technique to ensure that already memorized tasks do not impede learning.

Algorithm 1 Task-Informed Meta-Learning

Require: $p(\mathcal{T})$: Distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize meta model parameters θ_m, task encoder parameters θ_e
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$ with task information t_i
  for all $\mathcal{T}_i, t_i$ **do**
    Generate task embeddings $\mu_i = f(t_i; \theta_e)$
    Evaluate $\nabla_{\theta_m} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_m}, \mu_i)$
    Compute adapted meta parameters with gradient descent: $\theta'_{m_i} \leftarrow \theta_m - \alpha \nabla_{\theta_m} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_m}, \mu_i)$
  end for
  Update $\theta_m \leftarrow \theta_m - \beta \nabla_{\theta_m} \Sigma_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_{m_i}}, \mu_i)$
  Update $\theta_e \leftarrow \theta_m - \beta \nabla_{\theta_e} \Sigma_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_{m_i}}, \mu_i)$
end while

Task encoder TIML modulates parameters in the meta-learner based on embeddings calculated using task information. We encode the task-specific information into a set of vectors, two for each hidden layer to be modulated in the meta-model, denoted $\boldsymbol{t}_\gamma^k$ and $\boldsymbol{t}_\beta^k$ for the kth layer. We use feature-wise linear modulation (FiLM (Perez et al., 2018)) to modulate the hidden vector outputs of the meta-model using these task embeddings. That is, given a hidden vector output $\boldsymbol{h}$, we compute the modulated hidden vector output using a linear transformation and pass the modulated output to the next layer in the network: $\boldsymbol{h}_{out}^k = (\boldsymbol{t}_\gamma^k \odot \boldsymbol{h}^k) + \boldsymbol{t}_\beta^k$, where $\odot$ is the Hadamard product. Note that the task embeddings are updated in the outer loop during MAML training, so they remain constant for all datapoints when the meta-learner is being fine-tuned for a specific task (inner loop training).

We use a task encoder to learn the embeddings. This encoder consists of linear layers with GeLU activation (Hendrycks & Gimpel, 2016), group normalization (Wu & He, 2018) and dropout (Srivastava et al., 2014). The task information is encoded into a hidden task vector. Independent linear layers are then used to generate an embedding for each hidden vector in the classifier to be modulated. We keep the task encoder hyperparameters constant for all experiments (details in Appendix A), which demonstrates the insensitivity of this method to hyperparameter settings.

Forgetfulness We find that when the distribution of tasks is imbalanced, the model is likely to memorize over-represented tasks to the detriment of its ability to learn more difficult or rarer tasks (Collins et al., 2020). Unlike more complex methods designed to deal with this problem (Jamal & Qi, 2019; Collins et al., 2020; Baik et al., 2020b; Lee et al., 2019), explicit task information allows us to introduce a simple method to prevent memorization of certain tasks: removing training tasks the model has memorized. We define memorization as having exceeded a performance threshold for a task over a continuous set of epochs. We call this method "forgetfulness." Reducing the training set size before training has been previously explored (Ohno-Machado et al., 1998; Han et al., 2021) to reduce training time; forgetfulness does this dynamically to improve performance.

Table 1: Results for the **crop type classification** evaluation tasks. All results are averaged from 10 runs and reported with the accompanying standard error. We report the area under the receiver operating characteristic curve (AUC ROC) and the F1 score using a threshold of 0.5 to classify a prediction as the positive or negative class. We highlight the **first** and **second** best metrics for each task. TIML achieves the highest F1 score of any model on the Brazil task and the best AUC ROC and F1 scores when averaged across the 3 tasks. We highlight the improvement of TIML relative to other transfer/meta-learning methods, showing its ability to leverage task metadata when learning.

	Model	Kenya	Brazil	Togo	Mean
AUC ROC	Random Forest	0.578 ± 0.006	0.941 ± 0.004	0.892 ± 0.001	0.803
	No pre-training	0.329 ± 0.011	0.898 ± 0.010	0.861 ± 0.002	0.700
	Crop pre-training	0.694 ± 0.001	0.820 ± 0.002	$\mathbf{0.894 \pm 0.000}$	0.801
	MAML	$\mathbf{0.729 \pm 0.001}$	0.831 ± 0.005	0.878 ± 0.001	0.813
	MMAML (Vuorio et al., 2019)	0.690 ± 0.023	0.854 ± 0.037	0.878 ± 0.005	0.807
	TIML	$\mathbf{0.794 \pm 0.003}$	$\mathbf{0.988 \pm 0.001}$	0.890 ± 0.000	**0.890**
	no forgetfulness	0.779 ± 0.003	0.877 ± 0.003	0.893 ± 0.001	0.850
	no encoder	0.712 ± 0.001	$\mathbf{0.977 \pm 0.002}$	$\mathbf{0.895 \pm 0.000}$	**0.862**
	no task info or encoder	0.690 ± 0.001	0.977 ± 0.002	0.876 ± 0.001	0.848
F1 score	Random Forest	0.559 ± 0.003	0.000 ± 0.000	$\mathbf{0.756 \pm 0.002}$	0.441
	No pre-training	0.782 ± 0.000	$\mathbf{0.764 \pm 0.012}$	0.720 ± 0.005	0.734
	Crop pre-training	0.819 ± 0.001	0.619 ± 0.005	0.713 ± 0.002	0.613
	MAML	0.828 ± 0.001	0.496 ± 0.001	0.662 ± 0.001	0.652
	MMAML (Vuorio et al., 2019)	0.794 ± 0.006	0.720 ± 0.044	0.733 ± 0.007	**0.749**
	TIML	$\mathbf{0.838 \pm 0.000}$	$\mathbf{0.835 \pm 0.012}$	0.732 ± 0.002	**0.802**
	no forgetfulness	$\mathbf{0.840 \pm 0.000}$	0.537 ± 0.002	$\mathbf{0.764 \pm 0.002}$	0.724
	no encoder	$\mathbf{0.840 \pm 0.000}$	0.473 ± 0.002	0.691 ± 0.001	0.691
	no task info or encoder	0.837 ± 0.001	0.473 ± 0.001	0.645 ± 0.002	0.652

4 Datasets & Experimental Setup

We evaluated TIML on a range of datasets with heterogeneous tasks and limited available data but potentially useful task metadata. Specifically, we consider crop type prediction (a classification task) and crop yield estimation (a regression task) to demonstrate the suitability of TIML in both contexts. We also evaluated the performance of TIML on the Omniglot dataset.

4.1 Crop Type Classification

Up-to-date crop maps are critical to understanding the agricultural impacts of weather events and climate change (Song et al., 2021). Crop type classification – used to produce these maps – involves predicting whether or not a given instance contains a crop of interest. Specifically, given a remote sensing-derived pixel time series for a specific latitude and longitude and a crop of interest, the goal is to output a binary value describing whether the crop of interest is being grown at that location.

We use the CropHarvest dataset (Tseng et al., 2021b). This dataset, collected by NASA to inform agricultural policy and recently published in NeurIPS, consists of 90,480 globally distributed datapoints, of which 30,899 (34.2%) have multi-class agricultural labels and the remainder have binary "crop" or "non-crop" labels. Each datapoint is accompanied by a remotely sensed pixel time series from: Sentinel-2 L1C optical and Sentinel 1 synthetic aperture radar satellite observations, ERA5 climatology data (precipitation and temperature), and slope and elevation from a Digital Elevation Model. The time series includes one year of data at monthly timesteps.

As in the CropHarvest benchmarks, we constructed meta-learning tasks spatially using bounding boxes for countries drawn by Natural Earth (Patterson & Kelso). Tasks consist of binary classification of pixels as either crop vs. non-crop or a specific crop type vs. rest. This yielded 525 tasks, which were randomly split into training and validation tasks. We withheld the three CropHarvest evaluation tasks (described in Section 4.1) from the initial training. For each evaluation task, we fine-tuned the model on that task's training data before evaluating on that task's test data.

Task Metadata Task metadata is encoded in a 13-dimensional vector. Three dimensions encode spatial information, consisting of spherical latitude and longitude coordinates transformed to Cartesian coordinates (thus ensuring transformed values at the extreme longitudes are close to each other) using $[\cos(\text{lat}) \times \cos(\text{lon}), \cos(\text{lat}) \times \sin(\text{lon}), \sin(\text{lat})]$. The remaining 10 dimensions communicate the type of task the model is being asked to learn. This consists of a one-hot encoding of crop categories from the UN Food and Agriculture Organization (FAO) indicative crop classification (fao, 2020), with an added class for non-crop. For crop vs. non-crop tasks, positive examples are given the value $\frac{1}{n}$ across all the $n = 9$ crop type categories.

4.1.1 EVALUATION

The CropHarvest dataset includes 3 evaluation tasks that test the ability of a pre-trained model to learn from few in-distribution datapoints in a variety of agroecologies:

Togo crop vs. non-crop: The goal of this task is to classify datapoints as crop or non-crop in Togo. The training set consists of 1,319 datapoints and the test set consists of 306 datapoints – 106 (35%) positive and 200 (65%) negative – sampled from random locations within the country.

The two other evaluation tasks consist of classifying a specific crop. Thus, "rest" below includes all other crop and non-crop classes. For both tasks, entire polygons delineating a field (as opposed to single pixels within a field) were collected, allowing evaluation across the polygons. However, during training, only the polygon centroids were used.

Kenya maize vs. rest: The training set consists of 1,345 (266 positive and 1,079 negative) samples. The test set consists of 45 polygons with 575 (64%) positive and 323 (36%) negative pixels.

Brazil coffee vs. rest: The training set consists of 794 (21 positive and 773 negative) samples. The test set consists of 66 polygons with 174,026 (25%) positive and 508,533 (75%) negative pixels.

4.1.2 EXPERIMENTS

We evaluated TIML by training it on the CropHarvest dataset and fine-tuning it on the evaluation tasks, as was done for the benchmark results released with the dataset in (Tseng et al., 2021b). TIML can be applied to any neural network architecture. We use the same base classifier and hyperparameters as in (Tseng et al., 2021b): an LSTM model followed by a linear classifier.

Ablations We performed 3 ablations to quantify the contribution from the different components of TIML:

- **No forgetfulness**: TIML trained without forgetfulness; no tasks are removed in the training loop.
- **No encoder**: TIML with no encoder. The task information is instead appended to every raw input timestep and passed directly to the classifier.
- **No task information or encoder**: No task information passed to the model at all. This model is effectively a normal MAML model, trained with forgetfulness.

Baselines We compared TIML to 5 baselines. As with TIML, we fine-tuned these models on each benchmark task's training data and then evaluated them on the task's test data:

- **MMAML** (Vuorio et al., 2019): Multimodal Modal-Agnostic Meta-Learning, which infers task-clusters from the fine-tuning data and uses this to condition the MAML model.
- **MAML**: A Model-Agnostic Meta-Learning classifier without the task information.
- **Crop pre-training**: A classifier pre-trained to classify all data as crop or non-crop (without task metadata), then fine-tuned on the test task training data.
- **No pre-training**: A randomly initialized classifier, which is not pre-trained on the global CropHarvest dataset but instead is trained directly on the test task training data.

In addition, we trained a **Random Forest** baseline implemented using scikit-learn (Pedregosa et al., 2011) (further implementation details are available in Appendix C).

4.2 YIELD ESTIMATION

Accurate and timely yield estimates are a key input to food security forecasts (Becker-Reshef et al., 2020) and are necessary to better understand how food production can be sustainably managed (Lark et al., 2020). Yield estimation is a regression task which consists of predicting the amount of crop harvested per unit of land in a given area, given remote sensing data of that area. We estimate soybean yield in the highest soybean-producing states in the United States.

We recreated the yield prediction dataset originally collected by (You et al., 2017). This dataset consists of county-level soybean yields for the 11 U.S. states accounting for over 75% of national soybean production from 2009 to 2015 (shared under the U.S. Public Domain), and remote sensing data (specifically MODIS (Wan et al., 2015; Vermote, 2015) products, which are shared though the LP DAAC[1]). Since counties cover large areas, inputting the raw satellite data to the model would create extremely high-dimensional inputs. You et al. (2017) therefore assumed *permutation invariance*; that the positions of farmland pixels in a county do not affect yield, since they only indicate the positions of cropland. This allows all cropland pixels (selected using the MODIS land cover map (Friedl et al., 2010)) in a county to be mapped to a histogram of pixel values, significantly reducing the dimensionality of the input. We constructed meta-learning tasks by defining tasks as individual counties, with task $(\boldsymbol{X}, y)$ pairs consisting of histograms and yields for different years.

Task Metadata As in Section 4.1, we included the Cartesian-coordinate location of each task's county. We additionally included a one-hot encoding of which U.S. state the county is in.

Evaluation We used temporal validation to evaluate model performance: for each year in $\{2011, 2012, 2013, 2014, 2015\}$, we trained a model using all the data prior to that year, and evaluated the performance of the model for that year.

4.2.1 EXPERIMENTS

We applied TIML to the network architectures originally used by (You et al., 2017) – an LSTM and a CNN-based regressor. In addition to the remote sensing input, the Deep Gaussian Process baseline model (described below) receives as input the year of each training point. We therefore appended the year to each timestep of the input to the TIML LSTMs, so the model has comparable inputs to the Deep Gaussian Process. The CNN models receive only the remote sensing data as input.

Baselines We compared TIML to 2 baselines: the Deep Gaussian Process models (proposed by You et al. (2017) with the yield estimation dataset) and standard MAML. To train a Deep Gaussian Process, a deep learning model is first trained to estimate yield given the remote sensing dataset described above. The final hidden vector $h(x)$ of the model (for each input) is used as input to a Gaussian process $y(x) = f(x) + h(x)^T$ where $f(x) \sim \mathcal{GP}(0, k(x, x'))$. The kernel function k is conditioned on both the location of the datapoint (defined by its latitude and longitude) and the year of the datapoint. We included baselines with and without a Gaussian process (i.e., using the outputs of the deep learning models directly instead of passing the final hidden vectors to a Gaussian process). We note that this implementation of Deep Gaussian Processes by (You et al., 2017) differs from (Damianou & Lawrence, 2013). Finally, we highlight that the MAML LSTM model also receives the year appended as input, as is the case for the TIML LSTM model.

The MODIS datasets have been updated from version 5 to 6 since the original Deep Gaussian Process models were run. We therefore retrained the models to obtain our baseline results. We used the same hyperparameters as (You et al., 2017), with the addition of early stopping during training. We included the original results from (You et al., 2017) for comparison.

4.3 GROUPED-OMNIGLOT

The Omniglot dataset (Lake et al., 2015) is a one-shot learning dataset consisting of 1,623 handwritten characters drawn from 50 alphabets. We constructed tasks by considering only characters from a single alphabet together – a task therefore consists of one-shot classification of *characters drawn*

[1]All LP DAAC current data and products acquired through the LP DAAC have no restrictions on reuse, sale, or redistribution.

Table 2: The RMSE of county-level model performance for the **yield estimation** task. We use temporal validation to evaluate the model. Specifically, for each year, models are trained with data up to that year and evaluated with that year's data. All models are calculated from an average of 10 runs, with the standard error reported. We highlight the **first** and **second** best metrics for each task. For completeness, we include the results reported by (You et al., 2017), but highlight that these results were obtained on the MODIS 5.1 dataset (whilst all other models were trained on the MODIS 6.0 dataset) and are the result of 2 runs, compared to 10 runs for all other models. TIML improves on the Deep Gaussian Process models for both architectures, even though MAML performs significantly worse than other models. This suggests that in some cases, the task information is necessary for meta-learning to work.

Model	2011	2012	2013	2014	2015	Mean
LSTM	5.62 ± 0.10	6.60 ± 0.29	5.57 ± 0.21	6.63 ± 0.13	6.69 ± 0.31	6.22
+ GP	5.32 ± 0.10	$\mathbf{5.83 \pm 0.18}$	5.70 ± 0.19	5.61 ± 0.12	$\mathbf{5.24 \pm 0.14}$	**5.54**
+ MAML	26.90 ± 0.01	30.97 ± 0.01	29.57 ± 0.01	30.84 ± 0.01	32.02 ± 0.01	30.06
+ TIML	$\mathbf{5.16 \pm 0.03}$	$\mathbf{5.77 \pm 0.05}$	$\mathbf{5.39 \pm 0.02}$	5.24 ± 0.04	$\mathbf{4.89 \pm 0.04}$	**5.29**
CNN	6.08 ± 0.77	6.94 ± 1.83	6.42 ± 1.23	$\mathbf{4.80 \pm 0.83}$	5.57 ± 0.38	5.96
+ GP	5.55 ± 0.14	6.18 ± 0.49	6.44 ± 0.67	$\mathbf{4.87 \pm 0.31}$	6.02 ± 0.26	5.81
+ MAML	12.93 ± 0.05	8.28 ± 0.07	7.98 ± 0.04	12.05 ± 0.05	7.69 ± 0.06	9.79
+ TIML	$\mathbf{5.23 \pm 0.02}$	6.59 ± 0.02	$\mathbf{5.34 \pm 0.01}$	4.93 ± 0.02	6.35 ± 0.01	5.69
(You et al., 2017)						
LSTM + GP	5.77	6.23	5.96	5.70	5.49	5.83
CNN + GP	5.70	5.68	5.83	4.89	5.67	5.55

from the same alphabet. We highlight that this is a much more challenging setup than the typical setup of mixing all characters together, since more similar characters will need to be differentiated. We used a 5-way 1-shot regime to train and evaluate the model.

Task Metadata As task metadata, we used a one-hot encoding the alphabet from which a task is drawn. This metadata is less detailed than the crop classification and yield estimation tasks and is intended to demonstrate the utility of TIML even in scenarios with minimal metadata.

Evaluation For evaluation, we selected 5 characters per alphabet and held them out from the training set. We evaluated the models by fine-tuning them on a single example (per alphabet-set), and measuring the model accuracy across all remaining examples per character.

4.3.1 Experiments

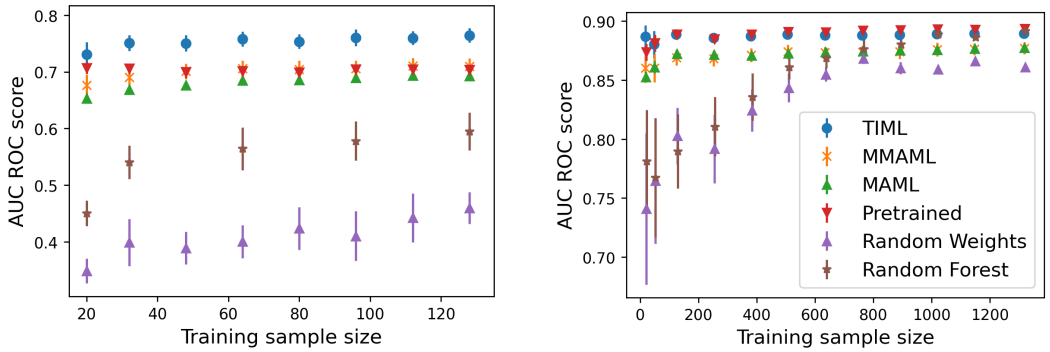
We evaluated TIML using the CNN architecture proposed for Omniglot and trained the model for $60,000$ steps as in Finn et al. (2017). As **baselines**, we compared TIML to MAML and MMAML (Vuorio et al., 2019), which we trained using the same model and training procedure. We emphasize that the Omniglot dataset is one of the datasets originally used by MAML Finn et al. (2017) and MMAML (Vuorio et al., 2019).

5 Results

5.1 Crop Type Classification

Table 1 shows the model results for TIML, its ablations and all baseline models when trained on the CropHarvest dataset. Like Tseng et al. (2021b), we report the AUC ROC score and the F1 score calculated using a threshold of 0.5. Overall, TIML is the best performing algorithm on the CropHarvest dataset, achieving the highest mean F1 and AUC ROC scores. TIML is consistently the best performing algorithm on every task.

TIML excels at learning from small dataset sizes. It is the only transfer/meta-learning model that outperforms a randomly-initialized model in the challenging Brazil task, where there are only 26 positive datapoints. We plot the performance of the models as a function of training set size in



(a) Kenya: Maize vs. Rest

(b) Togo: Crop vs. Non Crop

Figure 1: Results of TIML and the benchmark models when trained on a subset of the evaluation training data for the Crop Type Classification Task. We plot results for (Figure 1a) the Kenya Maize vs. rest evaluation task and (Figure 1b) Togo Crop vs. Non Crop evaluation task. Results are averaged from 10 runs and reported with standard error bars. Subsets are balanced so that they contain an equal number of positive and negative samples. For all training set sizes, TIML is the best performing model in the Kenya task and at or near best performance in the Togo task. We highlight that the advantage of TIML over other algorithms in general increases for smaller training sets.

Figure 1 for the Kenya and Togo evaluation tasks (the Brazil task is already in the small-dataset size regime). In both the Kenya and Togo tasks, TIML achieves the highest or near-highest ROC AUC scores for all subset sizes, and its advantage over other algorithms increases for smaller sample sizes.

5.2 Yield Estimation

We report the results for the yield estimation dataset in Table 2. Like You et al. (2017), we report the RMSE score averaged across all counties and use temporal validation to evaluate the models. The LSTM models used in TIML and MAML receive the year as input (to match the data provided to the Deep Gaussian Process), but the CNN models do not.

For both the LSTM and CNN architectures, TIML is the most performant model. This is the case even though the Deep Gaussian Process is much more memory-intensive, since it requires all predictions and hidden vectors (for the training and test data) to be computed together for the Gaussian process modelling step; this may be infeasible for larger datasets. TIML requires substantially less memory since it considers each county independently. It is also worth noting that while TIML achieves the best result of all models, MAML performs significantly worse than all other models. This suggests that in some contexts, the task metadata is necessary for meta-learning to work.

5.3 Grouped Omniglot

The results on grouped-omniglot for the 5-way 1-shot task are shown in Table 3. We re-emphasize the difficulty of this setup (which groups similar characters together) compared to usual (Finn et al. (2017)) approach of mixing all the characters, reflected in significantly poorer performance for the MAML baseline relative to this usual regime. The task metadata in this case is minimal, consisting of a one-hot encoding representing the alphabet a character-set was drawn from. Nonetheless, we find that TIML improves on both MAML and MMAML, indicating its effectiveness even with relatively little metadata.

6 Discussion

6.1 Importance of metadata encoding

The success of TIML is due not just to the presence of task metadata but also to the way that metadata is encoded and passed to the meta-learner. In the "no encoder" ablation conducted on the **crop type**

	Accuracy (%)
MAML	80.93 ± 1.06
MMAML	81.53 ± 1.03
TIML	$\mathbf{82.89 \pm 0.98}$

Table 3: Grouped-omniglot results, averaged from 3 random seeds with standard error, with the best results highlighted. TIML improves on both MAML and MMAML.

classification task, we provide task metadata directly to the learner by concatenating it with the input data. This approach does improve somewhat upon standard MAML, indicating the helpfulness of task metadata in learning, but it performs significantly worse than the full TIML algorithm.

6.2 Importance of forgetfulness

On the **crop type classification** task, we observe that using standard MAML or pre-training using global crop data actually results in lower performance on the Brazil task compared to an LSTM initialized with random weights. We hypothesize this may be due to the difference in distribution of the Brazil task data relative to the other tasks the models are trained on.

TIML, by contrast, performs significantly better than a randomly initialized LSTM, and indeed much better than all other methods. We see that forgetfulness plays a key role here, as training TIML without forgetfulness results in similar performance to the randomly initialized LSTM. However, it is not just forgetfulness that is key here, as training TIML with forgetfulness but without the metadata encoder causes the F1 score to drop precipitously. We therefore hypothesize that task information provides useful context around which tasks are being kept and forgotten during training, allowing TIML to learn from more difficult tasks in the "forgetful" regime without forgetting easier tasks it has already learned. Training TIML with forgetfulness significantly boosts performance in the Brazil task without substantially impacting performance on the other tasks, and yields significantly higher mean F1 and AUC ROC scores when measured across all tasks.

6.3 Task-explicit vs. task-inferred modulation

On the **grouped-omniglot** and **crop type classification** tasks, we compare the effect of passing explicit task information via TIML and of inferring this task information via MMAML (Vuorio et al., 2019). Overall, we find that when task metadata is present, it can lead to a significant improvement in performance compared to inferring task clusters.

We again highlight that this improvement specifically comes when the task information is added using TIML (Section 6.1). To our knowledge, TIML is the first approach that aims to leverage explicit metadata in gradient-based meta-learning algorithms.

7 Conclusion

We introduce task-informed meta-learning (TIML), a method for conditioning meta-learning models with explicit task metadata. The metadata is encoded into a set of vectors which are used to modulate the weights learned by a MAML learner prior to task-specific fine-tuning. TIML also includes a new technique called "forgetfulness," which we show can improve performance when there are many similar tasks to learn from. We evaluated TIML for both classification and regression tasks using a variety of neural network architectures (recurrent and convolutional networks), demonstrating its utility in a variety of regimes—including those with very few data points and those for which standard MAML fails completely. While usefulness for societal impact and geographic equity motivated us to focus in particular on agriculture-related tasks, we showed that TIML is not specific to agriculture and can be useful in any meta-learning problem with task-level metadata.

8 Reproducibility Statement

- **CropHarvest** is shared under a CC-BY-SA 4.0 license, and is available on Zenodo and via a python package. All code used to construct the dataset is currently open sourced at `github.com/nasaharvest/cropharvest`.

- The **yield estimation** dataset is constructed using Google Earth Engine (Gorelick et al., 2017). The code used to construct the data is open sourced, and the histograms used will be made available upon publication.
- The **Omniglot** dataset is open sourced and available at `github.com/brendanlake/omniglot`.

All code and models will be made available upon publication. In addition, implementation details are covered in depth in Appendix A.

REFERENCES

Programme, concepts and definitions. In *World Programme for the Census of Agriculture*. FAO, 2020.

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your MAML. In *International Conference on Learning Representations (ICML)*, 2019.

Sébastien M R Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for Meta-Learning research. 2020.

Kumar Ayush, Burak Uzkent, Chenlin Meng, Kumar Tanmay, Marshall Burke, David Lobell, and Stefano Ermon. Geography-aware self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-learning with adaptive hyperparameters. *Advances in Neural Information Processing Systems*, 2020a.

Sungyong Baik, Seokil Hong, and Kyoung Mu Lee. Learning to forget for meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020b.

Sungyong Baik, Janghoon Choi, Heewon Kim, Dohee Cho, Jaesik Min, and Kyoung Mu Lee. Meta-learning with task-adaptive loss function for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Inbal Becker-Reshef, Christina Jade Justice, Brian Barker, Michael Laurence Humber, Felix Rembold, Rogerio Bonifacio, Mario Zappacosta, Mike Budde, Tamuka Magadzire, Chris Shitote, Jonathan Pound, Alessandro Constantino, Catherine Nakalembe, Kenneth Mwangi, Shinichi Sobue, Terence Newby, Alyssa Whitcraft, Ian Jarvis, and James Verdin. Strengthening agricultural decisions in countries at risk of food insecurity: The GEOGLAM crop monitor for early warning. *Remote Sensing of Environment*, 2020.

Tony Chang, Brandon P Rasmussen, Brett G Dickson, and Luke J Zachmann. Chimera: A multi-task recurrent convolutional neural network for forest classification and structural estimation. *Remote Sensing*, 11(7):768, 2019.

Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. Task-robust model-agnostic meta-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Andreas Damianou and Neil D. Lawrence. Deep Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

Terrance de Vries, Ishan Misra, Changhan Wang, and Laurens van der Maaten. Does object recognition work for everyone? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

Mark A. Friedl, Damien Sulla-Menashe, Bin Tan, Annemarie Schneider, Navin Ramankutty, Adam Sibley, and Xiaoman Huang. MODIS collection 5 global land cover: Algorithm refinements and characterization of new datasets. *Remote Sensing of Environment*, 2010.

Noel Gorelick, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 2017.

Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and José MF Moura. Few-shot human motion prediction via meta-learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Rui Han, Chi Harold Liu, Shilin Li, Lydia Y. Chen, Guoren Wang, Jian Tang, and Jieping Ye. Slimml: Removing non-critical input data in large-scale iterative machine learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Neal Jean, Marshall Burke, Michael Xie, W. Matthew Davis, David B. Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 2016.

Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

Hannah Kerner, Gabriel Tseng, Inbal Becker-Reshef, Catherine Nakalembe, Brian Barker, Blake Munshell, Madhava Paliyam, and Mehdi Hosseini. Rapid response crop maps in data sparse regions. In *ACM SIGKDD Conference on Data Mining and Knowledge Discovery Workshops*, 2020.

Dan M Kluger, Sherrie Wang, and David B Lobell. Two shifts for crop mapping: Leveraging aggregate crop statistics to improve satellite-based maps in new regions. *Remote Sensing of Environment*, 2021.

Bernard Koch, Emily Denton, Alex Hanna, and Jacob Gates Foster. Reduced, reused and recycled: The life of a dataset in machine learning research. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015.

Tyler J. Lark, Seth A. Spawn, Matthew Bougie, and Holly K. Gibbs. Cropland expansion in the United States produces marginal yields at high costs to wildlife. *Nature Communications*, 2020.

Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *International Conference on Learning Representations*, 2019.

Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-Only Geographical Priors for Fine-Grained Image Classification. In *International Conference on Computer Vision (ICCV)*, 2019.

L. Ohno-Machado, H. S. Fraser, and A. Ohrn. Improving machine learning performance by removing redundant cases in medical data sets. *Proceedings. AMIA Symposium*, 1998.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Tom Patterson and Nathaniel Vaughn Kelso. Natural Earth. https://www.naturalearthdata.com/.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, 2019.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Marc Rußwurm, Sherrie Wang, Marco Korner, and David Lobell. Meta-learning for few-shot land cover classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020.

Andrei A Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2018.

Shreya Shankar, Yoni Halpern, Eric Breck, James Atwood, Jimbo Wilson, and D. Sculley. No classification without representation: Assessing geodiversity issues in open data sets for the developing world. In *NIPS 2017 workshop: Machine Learning for the Developing World*, 2017.

Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. On modulating the gradient for meta-learning. In *European Conference on Computer Vision*. Springer, 2020.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Xiao-Peng Song, Matthew C. Hansen, Peter Potapov, Bernard Adusei, Jeffrey Pickering, Marcos Adami, Andre Lima, Viviana Zalles, Stephen V. Stehman, Carlos M. Di Bella, Maria C. Conde, Esteban J. Copati, Lucas B. Fernandes, Andres Hernandez-Serna, Samuel M. Jantz, Amy H. Pickens, Svetlana Turubanova, and Alexandra Tyukavina. Massive soybean expansion in South America since 2000 and implications for conservation. *Nature Sustainability*, 2021.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 2014.

Sebastian Thrun and Lorien Pratt. *Learning to Learn*. Springer Science & Business Media, 1998.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations (ICLR)*, 2020.

Eleni Triantafillou, Hugo Larochelle, Richard Zemel, and Vincent Dumoulin. Learning a universal template for few-shot dataset generalization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Gabriel Tseng, Hannah Kerner, Catherine Nakalembe, and Inbal Becker-Reshef. Learning to predict crop type from heterogeneous sparse labels using meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021a.

Gabriel Tseng, Ivan Zvonkov, Catherine Nakalembe, and Hannah Kerner. CropHarvest: a global satellite dataset for crop type classification. In *Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021b.

Eric Vermote. MODIS/terra surface reflectance 8-day l3 global 500m SIN grid v006, 2015. URL `https://doi.org/10.5067/MODIS/MOD09A1.006`.

Risto Vuorio, Shao-Hua Sun, Hexiang Hu, and Joseph J. Lim. Multimodal model-agnostic meta-learning via task-aware modulation. In *Neural Information Processing Systems (NeurIPS)*, 2019.

Zhengming Wan, Simon Hook, and Glynn Hulley. MODIS/aqua land surface temperature/emissivity 8-day l3 global 1km SIN grid v006, 2015. URL `https://doi.org/10.5067/MODIS/MYD11A2.006`.

Anna X. Wang, Caelin Tran, Nikhil Desai, David Lobell, and Stefano Ermon. Deep transfer learning for crop yield prediction with remote sensing data. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS)*, 2018.

Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 2020.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, 2018.

Sang Michael Xie, Ananya Kumar, Robbie Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. In-n-out: Pre-training and self-training using auxiliary information for out-of-distribution robustness. In *International Conference on Learning Representations*, 2020.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *International Conference on Machine Learning*. PMLR, 2019.

Jiaxuan You, Xiaocheng Li, Melvin Low, David Lobell, and Stefano Ermon. Deep gaussian process for crop yield prediction based on remote sensing data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

A Implementation Details

We implement TIML in PyTorch (Paszke et al., 2019), using the learn2learn library (Arnold et al., 2020). All MAML and TIML models are trained using the same optimizer hyperparameters. Specifically, we use an inner loop learning rate of 10^{-4}. We use an Adam optimizer on the outer loop (for both the classifier and the encoder), with a Cosine Annealing Learning rate (as per (Antoniou et al., 2019)). For both the classifier and encoder, we use an initial learning rate of 10^{-4} and a minimum learning rate of 10^{-5}.

When fine-tuning, we use the same learning rate as the inner loop learning rate (10^{-4}) for all models with the exception of the i) yield-estimation standard-MAML CNN, for which we reduced the learning rate to 10^{-5} when fine-tuning it to handle issues with an exploding loss and ii) for the Omniglot models, for which we used an inner loop learning rate of 10^{-2} to reflect the values originally used in Finn et al. (2017).

Both MAML and TIML are trained for 1000 epochs – we selected the model checkpoint with the best performance on the validation set (consisting of 10% of the training tasks, up to a maximum of 50 tasks).

All TIML models were trained with the same **task encoder hyperparameters** (consisting of hidden blocks with sizes `[32, 64, 128]` and a dropout probability of `0.2`).

All models were trained on AWS. We used a `t2.xlarge` instance to train the LSTM moels, and a `p2.xlarge` instance to train the CNN models.

For the **crop type clasification** dataset, all LSTM-based classifiers were fine-tuned on the evaluation tasks for 250 gradient steps with batches containing 10 positive and 10 negative examples

(as in (Tseng et al., 2021b)). We show the variety of agro-ecologies represented in the crop type classification evaluation tasks in Figure 2.

For the **yield estimation** dataset, all models were fine-tuned on each county for 15 gradient steps, with batches of size 10. The reduced fine-tuning steps relative to the crop classification dataset is due to the much lower amount of data available for each county (compared to the crop classification evaluation tasks). Some counties did not have any fine-tuning data available – the results for these zero-shot counties are shared in Appendix B.

For the **Omniglot** dataset, models were finetuned for a single step to reflect the approach originally used in Finn et al. (2017).

A.1 FORGETFULNESS

We describe the thresholds used to define task memorization in Table 4.

Task	Metric	Threshold	Total Tasks	Removed Tasks
Crop Classification	AUC ROC ↑	0.95	463	141 (30%)
Yield Estimation	RMSE ↓	4	750	179 (24%)
Grouped-Omniglot	Accuracy ↑	0.99	50	4 (8%)

Table 4: The metrics and thresholds used to define task-memorization for each task, and the average number of tasks removed by the end of training. ↑ indicates that it is a lower threshold (we remove any task with an average metric above this threshold) while ↓ indicates an upper threshold (we remove any task with an average metric below this threshold).

For the crop-type and yield estimation experiemnts, a training task was forgotten if it met the threshold for forgetfulness continuously over the last 20 epochs. For the Omniglot datasets, the reduced tasks-per-epoch and increased variance per task (since any 5 characters in an alphabet could be used) motivated us to increase this lookback to 100 epochs. For the crop type classification, we note that the training batches were balanced to contain 10 positive and 10 negative examples, making AUC ROC an appropriate metric.

A.2 TASK AUGMENTATION FOR GEOSPATIAL MAML

Defining tasks according to their geospatial boundaries allows for a form of weak task augmentation, by including nearby datapoints which are not explicitly within the boundary. For example, using a rectangular bounding box instead of a polygon when defining a political boundary includes nearby points which may not be inside the polygon. Similarly, for the yield estimation dataset we include nearby counties in tasks for MAML and TIML.

B ZERO-SHOT LEARNING

For the **Yield estimation** task, some counties did not appear in the training data but were present in the evaluation data (i.e. if the first year of data for a county is 2011, then there will be no training data for that county for the evaluation year 2011).

For these counties, the model is therefore evaluated in a zero-shot learning regime (the county is not present when training the meta-model, or during fine-tuning).

We record the results of the yield model in a zero-shot learning regime below in Table 5. These results are included in the overall results reported in Table 2.

We highlight that very few counties are in this zero-shot regime, but include these results for completeness.

C RANDOM FOREST HYPERPARAMETERS

We consider two methods of hyperparameter selection for the random forest model:

(a) Togo (b) Kenya (c) Brazil

Figure 2: Example 1km $\times$ 1km satellite images of the evaluation regions, demonstrating the variety in field sizes and agroecologies being evaluated. (Images were obtained from Google Earth Pro basemaps comprised primarily of high resolution Maxar images, and are reproduced with permission from (Tseng et al., 2021b))

Model	2011	2012	2013	2014	2015
# counties	7	9	5	6	5
LSTM + TIML	8.99	12.93	17.19	9.97	11.22
CNN + TIML	10.44	7.02	9.81	7.25	11.89

Table 5: Zero-shot learning results: RMSE of the TIML model when measured only on counties not present during training (or fine-tuning). We note that these results were obtained with no training data about the county, in a zero-shot learning regime. The number of counties being tested is additionally recorded.

- Using the **default** hyperparameters which accompany the scikit-learn implementation.
- Conducting a **random grid search** with 5-fold cross validation. In this case, the hyperparameters are selected per randomly seeded run (i.e. different seeds of the same task may have different hyperparameters). We specifically conduct a grid search of the following hyperparameters and values: `"n_estimators": [10, 100, 200], "max_depth": [10, 50, None], "m_samples_leaf": [1, 2, 5]`. With a 5-fold cross validation, this trains 45 models per seed and selects the best performing set of hyperparameters.

The results of the tuned model (compared to the default implementation) are shown in Table 6, demonstrating the insensitivity of the random forest to hyperparameter tuning. Since the Random Forest with default hyperparameters obtains (slightly) better mean AUC ROC and F1 scores, we report these scores in the main paper.

We hypothesize that two factors drive this insensitivity: i) the small size of the evaluation tasks' training datasets, ii) the shift from points in the training sets to polygons in the test set (which better represent real world use of the model).

	Task	AUC ROC	F1
Tuned RF	Kenya	0.574 ± 0.015	0.536 ± 0.017
	Togo	0.895 ± 0.001	0.757 ± 0.002
	Brazil	0.921 ± 0.016	0.003 ± 0.002
	Mean	0.797	0.432
Default RF	Kenya	0.578 ± 0.006	0.559 ± 0.003
	Togo	0.892 ± 0.001	0.756 ± 0.002
	Brazil	0.941 ± 0.004	0.000 ± 0.000
	Mean	0.803	0.441

Table 6: The results of the tuned Random Forest and the Random Forest with the default hyperparameters.

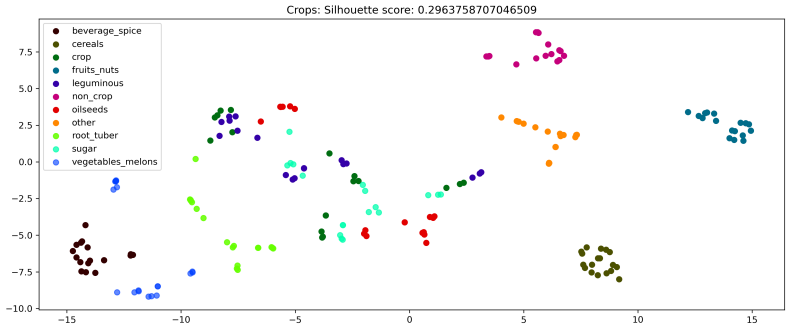


Figure 3: A plot of the FiLM (Perez et al., 2018) parameters for the crop-type classification task, reduced to 2 dimensions using t-SNE, coloured according to their crop label. We also included the Silhouette score of the embeddings in their original dimensions for reference. This shows strong clustering of certain classes (e.g. non-crop).

D FiLM PARAMETER CLUSTERS

We include a plot of FiLM (Perez et al., 2018) parameters in Figure 3, demonstrating the strong clustering for certain classes such as non-crop.