A. Implementation Details

We implement TIML in PyTorch [32], using the learn2learn library [3]. 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 [2]). 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. [13].

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** data set, 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 [48]). We show the variety of agroecologies represented in the crop type classification evaluation tasks in Figure 3.

For the **yield estimation** data set, 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 data set 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** data set, models were finetuned for a single step to reflect the approach originally used in [13].

A.1. Forgetfulness

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

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 data sets, 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) Togo (b) Kenya (c) Brazil

Figure 3. Example 1km × 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 [48])

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 data set 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:

- 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

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).

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.

	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.

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 data sets, ii) the shift from points in the training sets to polygons in the test set (which better represent real world use of the model).

D. FiLM parameter clusters

We include a plot of FiLM [35] parameters in Figure 4, demonstrating the strong clustering for certain classes such as non-crop.

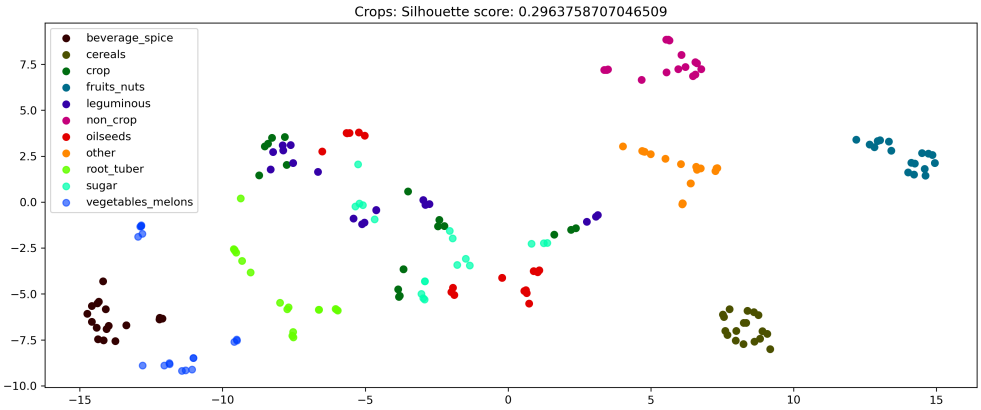

Figure 4. A plot of the FiLM [35] parameters for the crop-type classification task, reduced to 2 dimensions using t-SNE, coloured according to their crop label. We also included the Silhouette score of the embeddings in their original dimensions for reference. This shows strong clustering of certain classes (e.g. non-crop).