Reproducibility Review of: Generalizability of Foundation Models: A Case Study on Cocoa Mapping Across Countries Using Sparse Labels

Item	Value
Title	Generalizability of Foundation Models: A Case Study on Cocoa Mapping Across Countries Using Sparse Labels
Authors	Ruslan Mammadov, Paul Walther, Julius Fricke, Martin Werner
Ref. paper	[reference to the paper, DOI or URL]
Codechecker(s)	Joseph Shingleton
Date of Check	2026-03-31
Summary	Partial Reproduction
Repository	https://github.com/osapiens-Terra-GmbH/2026-AGILE-CocoaMapping.git.
Ref. certificate	https://doi.org/10.17605/OSF.IO/YH8F6

Table 1: Reproduction metadata

Summary

The paper is mostly reproducible. Some parts of the paper were not reproduced due to limited data accessibility and some missing code, however the results relating to the paper's central arguments are retrievable. There was some small delta between the reported and retrieved results, typically under 2%. This delta is due to data sharing limitations which are explained in the paper.

Reproducibility reviewer notes

Installation prerequisites and computational environment

Installation was relatively straight forward thanks to the provision of a `requirements.txt` file and clear setup instructions. I had to slightly adapt the requirements to suit my system (NVIDIA RTX 5090 with CUDA 12.8). In particular, I was unable to run without having the cu128 version of Pytorch installed. It is possible that the different CUDA version contributed to the small difference between observed and reported results.

Data preparation

Most of the data was made available via a temporary Google Drive folder – with some exceptions due to data sharing restrictions. This is a big dataset, and while google drive is fine for this, it does cause some problems. The very large `.tif` and `.hdf5` files can not be compressed by Google Drives folder downloader, and so get downloaded as separate files into the root target folder (i.e. `/downloads/`). Since many files in the repository are only differentiated by path rather than filename (e.g. `data/folder_1/val.hdf5` and `data/folder_2/val.hdf5`), it made it difficult to know where each file should go. This isn't a major issue; it just took a little trial and error to make sure everything was organized correctly.

Running the code

My initial run of the code was unsuccessful due to problems with the downloaded data. Deleting the data and rerunning with forced downloading rectified this error and allowed the code to run.

The code is arranged into a series of Python scripts. Running each individually provides a JSON structured output relating to some part of the paper. While the code is easy to run, it is not immediately obvious which outputs relate to which parts of the paper. This could be rectified by either adding more detail to the README.md or by running the scripts through a jupyter notebook with annotation.

Outputs and results

While the code (with modification) runs easily, it was not immediately clear which results correspond to particular parts of the manuscript. After discussion with the paper's author's the relevant results were identified and linked with the output results. While exact replication was not possible due to the limited public dataset, the obtained results were within a few percentage points of the published results and so are mostly reproducible.

Table 1-5 reproducibility

Tables 1-5 in the published paper report results obtained from prior studies, and so are not relevant to this reproducibility review.

Table 6 reproducibility

Table 6 (Zero-shot performance on the Nigeria set) can be reproduced with the public subset to within a few percentage points of the published data. Further, the overall ranking of the models is maintained, which is more relevant to the claims of the paper. The results are obtained by running scripts in `scripts/scripts_zero_shot_transfer/*.py`.

Table 6. Zero-shot performance on the Nigeria test set.

Model	F1 (%)	Rec. (%)	Prec. (%)
Baseline CNN	**61.0**	43.9	99.9
CROMA	56.3	39.3	99.4
CROMA (frozen enc.)	37.7	23.3	98.8
AlphaEarth	5.8	3.0	99.8
GCHM	51.5	34.7	99.8

Table 1: Reproduction of outputs detailed in table 6 of the submitted paper (above).

Model	F1	Δ F1	Rec	Δ Rec	Prec	Δ Prec	Rank	Δ Rank
Baseline CNN	62.0	+1.0	45.0	+1.1	99.9	+0.0	1	=
CROMA	54.0	-2.3	36.2	-3.1	99.4	+0.0	2	=
CROMA (frozen enc.)	36.1	-1.6	22.1	-1.2	98.6	-0.2	4	=
AlphaEarth	5.8	+0.0	3.0	+0.0	99.8	+0.0	5	=
GCHM	53.2	+1.7	36.2	+1.5	99.9	+0.1	3	=

Figure 7 and section 5.1 reproducibility

Table 7 can not be reproduced due to limited data availability. The results detailed in Figure 7, and the corresponding text in section 5.1, are mostly reproducible using the available data and code. There is no clear method within the code by which the full validation curves in figure 7 can be reproduced, however the final validation values after

100 epochs are mostly reproducible. The results are obtained by running the scripts in `scripts/scripts_source_region_val/*.py`.

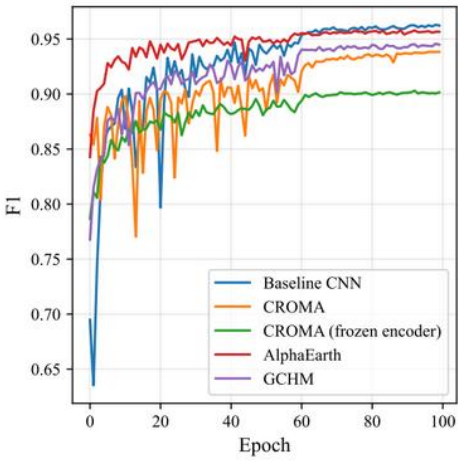

Figure 7. F1 validation scores for the main models on the source region per training epoch. The baseline CNN starts with the lowest F1 but surpasses all other models after roughly the 60th epoch.

> Validation curves based on 100,000 samples (Figure 7) show a consistent ordering: the baseline CNN achieves the highest validation F1 (96.3%), followed by AlphaEarth (95.7%), GCHM (94.6%), CROMA (93.8%), and frozen CROMA (90%). The baseline CNN starts with the lowest F1 but surpasses all pre-trained models after about 60 epochs.

Figure 1: Text from section 5.1 detailing the final validation F1 scores over the training period.

Table 2: Reproduction of the final validation F1 scores detailed in figure 7 and section 5.1 of the paper (above).

Model	F1	Δ F1	Rank	Δ Rank
Baseline CNN	95.7	+0.0	1	=
CROMA	93.4	-0.4	4	=
CROMA (frozen enc.)	89.5	-0.5	5	=
AlphaEarth	95.5	-0.2	2	=
GCHM	94.0	-0.6	3	=

Table 8 reproducibility

Table 8 is mostly reproducible, although again the limited dataset means that exact results are not retrievable. The results are obtained by running the scripts in `scripts/scripts_nigeria_finetuned/*source_train_finetuned.py`. The retrieved results are summarized in table 3.

Table 8. Performance on the Nigeria test set after fine-tuning with sparse labels.

Model	F1 (%)	Rec. (%)	Prec. (%)
Random Initalization			
Baseline CNN	46.8	30.1	96.8
GCHM	72.7	57.7	98.2
Trained on source			
Baseline CNN	88.4	79.7	99.1
CROMA	**95.1**	92.4	98.0
CROMA (frozen enc.)	90.6	84.0	98.2
AlphaEarth	**95.7**	95.1	96.3
GCHM	89.1	80.8	99.4

Table 3: Results obtained from the provided data and code, corresponding to table 8 in the paper (above).

Model	F1	Δ F1	Rec	Δ Rec	Prec	Δ Prec	F1 Rank	Δ F1 Rank
Random initialization								
Baseline CNN	46.7	+0.1	30.8	+0.7	96.8	+0.0	7	=
GCHM	73.0	+0.3	58.0	+0.3	98.3	+0.1	6	=
Trained on source								
Baseline CNN	89.4	+1.0	80.9	+1.2	99.8	+0.7	4	-1
CROMA	96.3	+1.2	93.2	+0.8	99.6	+1.6	2	=
CROMA (frozen enc.)	91.2	+0.6	84.2	+0.2	99.5	+1.3	3	=
AlphaEarth	97.4	+1.7	95.1	+0.0	99.9	+3.6	1	=
GCHM	88.6	-0.5	79.8	-1.0	99.5	+0.1	5	+1

Table 9 reproducibility

Table 9 is mostly reproducible. The results where “Source Train” is “Yes” correspond to those reported for table 8. The remaining results (“Source Train” = “No”) are obtained by running the scripts in `scripts/scripts_nigeria_finetuned/*_no_source_finetuned.py`.

Table 9. Comparison of fine-tuning results with and without source-region training.

Model	Source Train.	F1 (%)	Rec. (%)	Prec. (%)
CROMA	Yes	95.1	92.4	98.0
CROMA	No	95.0	92.1	98.0
CROMA (frozen enc.)	Yes	90.6	84.0	98.2
CROMA (frozen enc.)	No	90.6	84.3	98.0
AlphaEarth	Yes	95.7	95.1	96.3
AlphaEarth	No	92.5	88.1	97.5
GCHM	Yes	89.1	80.8	99.4
GCHM	No	91.3	84.8	98.8

Table 4: Reproduction of the results for Table 9 where "Source Train." = "No" (above). See previous section for reproduction of other rows in the table.

Model	F1	Δ F1	Rec	Δ Rec	Prec	Δ Prec	F1 Rank	Δ F1 Rank
CROMA	94.9	-0.1	90.7	-1.4	99.5	+1.5	1	=
CROMA (frozen enc.)	91.8	+1.2	85.0	+0.7	99.7	+1.7	3	-1
AlphaEarth	93.7	+1.2	88.1	+0.0	100.0	+2.5	2	=
GCHM	91.2	-0.1	84.7	-0.1	98.9	+0.1	4	+1

Tables 10-14 reproducibility

Table 10 summarises and aggregates information from tables 8 and 9, and so is mostly reproducible using the methods describe above. The ablation studies detailed in tables 11-14 can not be tested given the current code and data set-up, and so are not reproducible. This should be stated explicitly in the README file of the reproducibility instructions, along with reasoning for their absence.

Figure reproducibility

There is no mechanism in the code for reproducing figures 7-9. While final F1 scores do correspond to some of the code outputs, the full validation curves can not be reproduced. This does not necessarily matter as they are incidental to the main conclusions of the paper, however for proper documentation of the scientific process it would have been helpful to include them in the reproducibility report.

Recommendations

The results in the paper were mostly reproducible, given the constraints of the data availability. I would advise the following for improved reproducibility:

1) Make sure that running the code produces the full figures where appropriate. This is particularly relevant for the validation curves – I was able to obtain the final results (within a small error), however it would be helpful if we could also produce the related plots and figures.

2) Make the connection between outputs and results more explicit. Notebooks are a good option for this as they allow you to annotate each step with markdown. It would be helpful if the outputs were structured to look more like the specific tables in the paper.

3) Be explicit about which parts of the paper are out-of-scope for review. There were many tables in the paper which were not reproducible, either due to lack of data availability or due to them relating to externally sourced figures. This should be made clear in the README of the repository.

Acknowledgements

Thanks to the paper's author for helpful collaboration on this review.

Citing this document

This report is part of the reproducibility review at the AGILE conference. For more information see https://reproducible-agile.github.io/. This document is published on OSF/ResearchEquals at https://doi.org/10.17605/OSF.IO/YH8F6 .

To cite the report use

https://doi.org/10.17605/OSF.IO/YH8F6