PRUE: A Practical Recipe for Field Boundary Segmentation at Scale

Supplementary Material

A. Extended Experimental Details

Details on instance and panoptic segmentation model baselines. Delineate Anything [37] is based on Ultralytics' YOLOv11-seg and fine-tuned on FBIS-22M, a dataset of RGB images from multiple remote sensing sources (Sentinel-2, Planet, Maxar, Pleiades, orthophotos) over 9 European countries with spatial resolution 0.25-10m. For Delineate Anything, we perform a 1st-99th percentile normalization following the pretraining dataset norms. SAM [35] is a promptable instance segmentation model pretrained on natural images, which we assessed in both zero-shot and fine-tuned settings as described in the methods section. Mask2Former (M2F) [8] is a universal segmentation architecture capable of semantic, instance, and panoptic segmentation depending on training configuration; we adapted M2F with a Swin-S backbone to handle the 8-channel RGBN bitemporal input and train it on the panoptic task, which jointly predicts individual field instances (things) and background land cover classes (stuff). Note that SAM and Mask2Former were trained without presence-only label masking–a data preprocessing strategy used by all semantic baselines that filters out ambiguous background regions in partially-labeled countries. The inability to implement this masking for instance models (due to fundamental differences in how instance segmentation models handle training objectives) means these models faced a harder optimization landscape, being penalized for predicting fields in regions that may contain unlabeled fields.

RGB-only comparison with Delineate Anything. To provide a fairer comparison with Delineate Anything (DelAny) [37], we trained new PRUE models with EfficientNet-B3 (EF3) backbones on RGB-only data from a single time step. This resulted in object F1 scores of 0.38 ± 0.06 (window A) and 0.37 ± 0.06 (window B), both higher than DelAny's performance despite using the same RGB-only input. This demonstrates that the performance gap between PRUE and DelAny is not attributable to FTW's additional spectral channel (NIR), but rather reflects differences in model design and training data diversity.

SAM2 evaluation. We evaluated SAM2 in a zero-shot setting on the window A RGB bands, resulting in a pixel IoU of 0.31 and an object F1 of 0.07. SAM2 is designed for video segmentation where it expects continuous video frames in which objects move or deform slightly with strong appearance consistency [47]. This differs fundamentally from the FTW setting, which provides two snapshots separated by months that capture significant phenological changes. This creates multiple failure modes: (1) appearance shifts between seasons violate SAM2's visual consistency assumption, and (2) fields do not "move" between timestamps—they transform—leaving SAM2's optical flow and correspondence mechanisms without useful signal.

Geospatial foundation models (GFMs). The FTW benchmark provides four bands (RGB and NIR) [34], which is fewer than the multi-spectral inputs used by most GFMs. Since many GFMs expect 8 to 13 Sentinel-2 bands, we used GFMs evaluation wrapper published in Galileo codebase [62] to correctly prepare inputs for Galileo [62], CROMA [25], SoftCon [68], Prithvi 2.0 [58], DOFA-v1 [73], DeCUR [67], and Satlas [4]. This wrapper allowed us to (1) construct the band set expected by each model (applying mask where applicable), (2) impute missing channels in a model-consistent manner, and (3) apply each model's required normalization or standardization using its original training statistics. For TerraFM [14], we assign zeros to all missing spectral bands. DINOv3 [57] operates exclusively on RGB inputs, so the FTW RGB bands are passed directly without modification. For Clay [9] and TerraMind [32], which are designed to handle partially missing spectral information, we provide the available four-band input with the appropriate normalization for each model. Patch-level embeddings are extracted from each pretrained GFM independently for the two temporal windows defined in FTW.

GFM feature fusion and decoding. The patch embeddings from the two temporal windows are fused by first concatenating them along the feature dimension and passing the result through a three-layer MLP. Our objective is to evaluate the representational quality of the frozen GFM features themselves. A common evaluation strategy adopted for this objective is linear probing. However, we argue that a single linear transformation is often too limited to fully assess the spatial and contextual information encoded in the pretrained features. Conversely, using a specialized model such as U-Net [50] at this stage would primarily test the ability of the decoder rather than the underlying GFM embeddings. To strike a balance between these two extremes, we employ a simple decoder that provides moderate flexibility through a 3×3 projection layer, two residual refinement blocks, and a multi-scale convolutional module, followed by pixel-shuffle upsampling. Table 1 reports the results obtained with our convolutional decoder, and Table 4 provides the complementary 1×1 convolution linear-probing results.

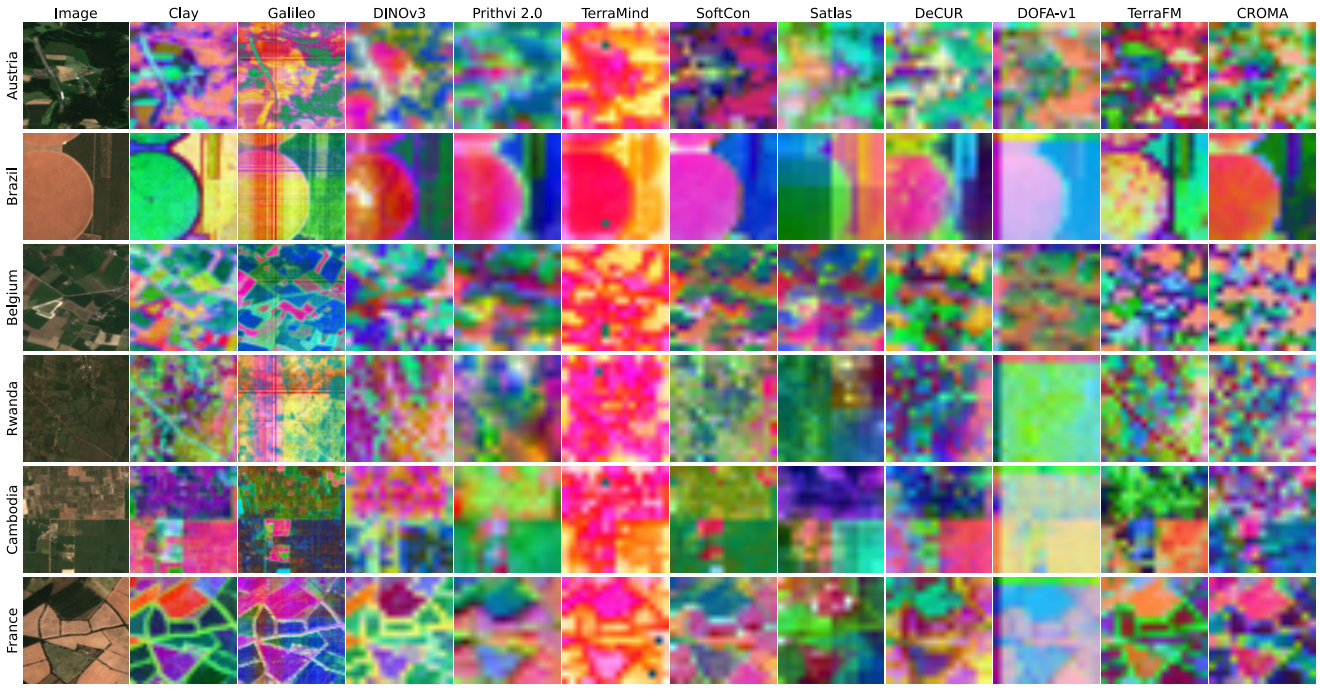

Figure 4. PCA visualization of frozen GFM encoder features for a representative subset of image examples. The top 3 principal components of patch embeddings are displayed as RGB channels. Clay (8×8 patch sizes) and Galileo (4×4 patch size) capture finer spatial structure and more distinct field boundaries compared to models using 16×16 patches, demonstrating how tokenization granularity affects feature quality for segmentation tasks. See Table 4 for quantitative performance.

Table 4. GFM linear probing results using a lightweight decoder (1×1 convolution + bilinear upsampling), sorted by object-level F1. Clay (ViT-Large, 8×8 patches) and Galileo (ViT-Base, 4×4 patches) outperform other GFMs that use coarser 16×16 patch sizes, due to the finer patch resolutions as well as techniques intentionally designed to handle missing spectral bands.

Model	**Backbone**	**Pixel-level**			**Object-level**				
		IoU ↑	Prec ↑	Recall ↑	Prec ↑	Recall ↑	F1 ↑	$AP_{0.5:0.95}$ ↑	$AP_{0.5}$ ↑
Clay	ViT-Large	**0.56**	0.88	**0.60**	0.22	0.16	**0.18**	0.07	0.17
Galileo	ViT-Base	0.53	0.83	0.59	0.11	**0.19**	0.13	**0.08**	**0.18**
DINOv3	ViT-Large	0.47	**0.89**	0.50	**0.25**	0.09	0.12	0.03	0.08
Prithvi 2.0	ViT-Large	0.44	0.84	0.48	0.20	0.06	0.10	0.02	0.06
TerraMind	ViT-Base	0.44	0.85	0.47	0.19	0.07	0.10	0.02	0.06
SoftCon	ViT-Small	0.41	0.83	0.46	0.16	0.05	0.07	0.01	0.04
Satlas	Swin-Tiny	0.39	0.74	0.45	0.13	0.04	0.07	0.01	0.03
DeCUR	ViT-Small	0.42	0.80	0.46	0.15	0.04	0.07	0.01	0.03
DOFA-v1	ViT-Large	0.39	0.77	0.44	0.14	0.04	0.06	0.01	0.03
TerraFM	ViT-Base	0.44	0.85	0.48	0.17	0.06	0.09	0.02	0.05
CROMA	ViT-Base	0.42	0.85	0.46	0.18	0.05	0.08	0.02	0.05

B. Extended Results

Our design choice is a result of extensive ablations. Table 5 shows that boundary weighting, loss function, and targeted augmentations have the strongest impact on performance. Moderate class weights ($\omega \approx 0.75$) and losses such as LogCosh Dice, Tversky, and Local Tversky consistently improve object F1 and pixel IoU, while brightness, scale, and channel-shuffle augmentations provide additional robustness. Larger EfficientNet backbones slightly enhance results, and combining these components in PRUE yields the highest accuracy and boundary agreement across all metrics.

Accuracy–throughput trade-off across model configurations. To explicitly summarize the accuracy–cost trade-off over all models, Figure 5 shows the Pareto front between object F1 and throughput. Supplemental Table 5 compares our methodology against the baseline using the same backbone (EfficientNet-B3): the "U-Net LogCosh Dice 0.75-weight Augs EN-B3" row shows that PRUE with an EF3 backbone achieves an object F1 of 0.43 ± 0.07 and field IoU of 0.74 ± 0.07, compared to PRUE-EF7 which achieves

Table 5. Ablation results for controlled experiments on the FTW test set (excluding presence-only countries) in which each row varies a single design choice (data augmentations, class weighting, encoder, loss function, or architecture). **Bold** indicates best performance, underline indicates second-best.

Category	Ablation	Performance		Input order		Brightness		Scale		Agree.
		Object F1 ↑	Pixel IoU ↑	F1|Δ| ↓	IoU|Δ| ↓	F1|Δ| ↓	IoU|Δ| ↓	F1|Δ| ↓	IoU|Δ| ↓	Avg↑
	FTW-v1	0.39 ± 0.08	0.68 ± 0.08	0.07	0.11	0.04	0.05	0.17	0.08	0.93
Data augs	Brightness	0.39 ± 0.08	0.68 ± 0.08	0.07	0.09	**0.00**	**0.00**	0.14	0.05	0.93
Data augs	Resize	0.38 ± 0.08	0.67 ± 0.09	0.07	0.10	0.04	0.05	**0.00**	0.01	0.94
Data augs	Brightness+Resize	0.38 ± 0.08	0.66 ± 0.09	0.06	0.10	0.03	0.03	**0.00**	0.02	0.95
Data augs	Channel shuffle	0.39 ± 0.07	0.68 ± 0.09	**0.00**	**0.00**	0.05	0.05	0.18	0.09	0.94
Class weights	$\omega = 0.60$	0.32 ± 0.06	0.76 ± 0.06	0.07	0.13	0.07	0.07	0.28	0.19	0.96
Class weights	$\omega = 0.65$	0.36 ± 0.06	0.76 ± 0.06	0.07	0.11	0.07	0.07	0.30	0.17	0.96
Class weights	$\omega = 0.70$	0.40 ± 0.06	0.75 ± 0.06	0.08	0.12	0.08	0.07	0.30	0.14	0.95
Class weights	$\omega = 0.75$	0.42 ± 0.06	0.74 ± 0.07	0.08	0.11	0.07	0.07	0.29	0.13	0.95
Class weights	$\omega = 0.80$	0.42 ± 0.07	0.73 ± 0.06	0.09	0.12	0.07	0.07	0.23	0.11	0.95
Class weights	$\omega = 0.85$	0.41 ± 0.07	0.70 ± 0.08	0.08	0.10	0.05	0.06	0.17	0.08	0.96
Encoder	EfficientNet-B4	0.40 ± 0.07	0.69 ± 0.09	0.07	0.09	0.04	0.05	0.15	0.06	0.93
Encoder	EfficientNet-B5	0.41 ± 0.07	0.70 ± 0.08	0.07	0.11	0.04	0.05	0.16	0.10	0.93
Encoder	EfficientNet-B6	0.41 ± 0.07	0.70 ± 0.08	0.07	0.11	0.04	0.05	0.18	0.14	0.94
Encoder	EfficientNet-B7	0.42 ± 0.07	0.71 ± 0.08	0.07	0.09	0.04	0.04	0.21	0.09	0.94
Encoder	MiT-B2	0.39 ± 0.08	0.67 ± 0.09	0.08	0.10	0.02	0.03	0.13	0.05	0.95
Encoder	MiT-B3	0.39 ± 0.08	0.67 ± 0.09	0.08	0.10	0.02	0.03	0.13	0.05	0.95
Encoder	MiT-B4	0.39 ± 0.08	0.68 ± 0.09	0.07	0.09	0.02	0.03	0.16	0.06	0.94
Encoder	MiT-B5	0.38 ± 0.08	0.67 ± 0.09	0.08	0.10	0.02	0.03	0.12	0.02	0.95
Encoder	ResNet-18	0.35 ± 0.07	0.67 ± 0.09	0.08	0.11	0.04	0.05	0.14	0.05	0.93
Encoder	VGG13-BN	0.38 ± 0.07	0.69 ± 0.08	0.08	0.10	0.04	0.05	0.27	0.20	0.94
Learning rate	0.0001	0.34 ± 0.07	0.65 ± 0.09	0.05	0.07	0.03	0.04	0.15	0.08	0.89
Learning rate	0.0003	0.37 ± 0.07	0.66 ± 0.09	0.06	0.08	0.04	0.04	0.17	0.10	0.90
Learning rate	0.003	0.39 ± 0.08	0.68 ± 0.09	0.08	0.10	0.06	0.08	0.17	0.08	0.95
Learning rate	0.01	0.39 ± 0.08	0.68 ± 0.09	0.08	0.11	0.06	0.06	0.18	0.08	0.95
Learning rate	0.03	0.37 ± 0.08	0.67 ± 0.09	0.09	0.10	0.10	0.13	0.16	0.07	0.94
Loss function	CE (w/ Edge Agreement)	0.39 ± 0.08	0.68 ± 0.09	0.07	0.10	0.04	0.05	0.15	0.07	0.94
Loss function	CE + Dice	0.41 ± 0.07	0.70 ± 0.08	0.07	0.10	0.04	0.05	0.20	0.08	0.93
Loss function	CE + Dice (no class weights)	0.38 ± 0.07	$\mathbf{0.77 \pm 0.06}$	0.08	0.11	0.06	0.05	0.29	0.15	0.95
Loss function	CE + FTNMT	0.41 ± 0.07	0.70 ± 0.08	0.08	0.11	0.04	0.05	0.21	0.08	0.94
Loss function	CE (no class weights)	0.24 ± 0.06	$\mathbf{0.77 \pm 0.06}$	0.05	0.11	0.05	0.06	0.15	0.14	0.96
Loss function	Dice	0.42 ± 0.07	0.76 ± 0.07	0.08	0.13	0.06	0.06	0.31	0.16	0.96
Loss function	Dice (w/ Edge Agreement)	0.42 ± 0.07	$\mathbf{0.77 \pm 0.06}$	0.08	0.13	0.07	0.07	0.32	0.17	0.95
Loss function	Focal	0.18 ± 0.05	0.75 ± 0.06	0.04	0.10	0.04	0.06	0.15	0.21	0.96
Loss function	FTNMT	0.38 ± 0.06	0.79 ± 0.06	0.10	0.14	0.06	0.07	0.29	0.17	0.93
Loss function	Local Tversky	0.45 ± 0.07	0.74 ± 0.07	0.10	0.13	0.05	0.05	0.31	0.16	0.94
Loss function	LogCosh Dice	0.44 ± 0.07	$\mathbf{0.77 \pm 0.06}$	0.09	0.13	0.06	0.06	0.33	0.20	0.94
Loss function	LogCosh Dice + CE	0.39 ± 0.08	0.68 ± 0.08	0.08	0.10	0.04	0.05	0.14	0.06	0.93
Loss function	Tversky	0.43 ± 0.07	0.76 ± 0.06	0.09	0.12	0.06	0.05	0.34	0.15	0.95
Loss function	Tversky + CE	0.41 ± 0.07	0.71 ± 0.08	0.07	0.10	0.05	0.06	0.20	0.10	0.92
Architecture	FCN	0.14 ± 0.03	0.60 ± 0.08	0.04	0.09	0.03	0.07	0.10	**0.00**	**0.99**
Architecture	FCSiam	0.40 ± 0.07	0.69 ± 0.08	**0.00**	**0.00**	0.05	0.06	0.23	0.10	0.92
Architecture	UNETR	0.37 ± 0.07	0.69 ± 0.08	0.08	0.10	0.04	0.04	0.27	0.18	0.94
Architecture	UPerNet	0.34 ± 0.08	0.64 ± 0.10	0.07	0.09	0.03	0.04	0.13	0.04	0.91
Combination	FCSiam Combo	0.44 ± 0.07	0.75 ± 0.07	**0.00**	**0.00**	0.04	0.04	0.05	0.02	0.94
Combination	U-Net LogCosh Dice Augs EN-B3	0.42 ± 0.07	0.74 ± 0.07	**0.00**	**0.00**	**0.00**	**0.00**	0.01	0.01	0.94
Combination	U-Net LogCosh Dice 0.75-weight Augs EN-B3	0.43 ± 0.07	0.74 ± 0.07	**0.00**	**0.00**	**0.00**	**0.00**	0.02	**0.00**	0.94
Combination	U-Net LogCosh Dice 0.75-weight Augs EN-B5	0.46 ± 0.07	0.75 ± 0.07	**0.00**	**0.00**	**0.00**	**0.00**	0.01	0.01	0.94
Combination	**PRUE**	$\mathbf{0.47 \pm 0.07}$	0.76 ± 0.08	**0.00**	**0.00**	**0.00**	**0.00**	0.01	0.01	0.95

0.47 ± 0.07 and 0.76 ± 0.08, respectively. PRUE-EF3 is a Pareto improvement over the FTW baseline, meaning it achieves higher accuracy *without* sacrificing throughput. The PRUE family forms a substantially stronger Pareto frontier than any prior model and shows a clear throughput vs. performance trade-off. As a practical reference, an EF3 backbone can process the entire land area of the Earth in approximately 66 hours on a single V100 GPU, while EF7 would require approximately 134 hours.

Full fine-tuning of Clay. To assess whether the performance gap between GFMs and PRUE could be closed with end-to-end training, we fully fine-tuned the best-performing GFM, Clay, selecting the learning rate from $\{1, 3\} \times 10^{\{-5,-4,-3\}}$ based on the best object F1 on the validation set. Full fine-tuning increased Clay's object F1 from 0.36 to 0.42 (see Figure 5) and pixel IoU from 0.67 to 0.73. While these represent meaningful improvements over frozen-encoder decoding, both metrics remain below PRUE (0.47 object F1, 0.76 pixel IoU), and Clay's throughput is substan-

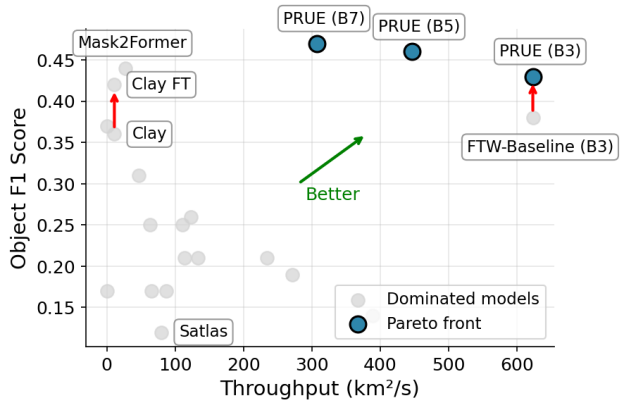

Figure 5. Pareto front between Object F1 and Throughput for all Table 1 models, including PRUE-EF (B3/B5) throughput. PRUE-EF3 is a Pareto improvement over the FTW baseline. The PRUE family forms a substantially stronger Pareto frontier than any prior model.

tially lower (Clay: 11 km^2/s vs. PRUE: 307 km^2/s). This confirms that the performance gap is not solely attributable to the frozen-encoder evaluation protocol, but reflects fundamental differences in spatial resolution and architectural suitability for field boundary delineation.

1D convolution did not capture field boundary complexity. Across all GFM experiments, we observed that decoding using a single 1×1 convolution followed by bilinear upsampling produced coarse, low-fidelity boundaries compared to our convolutional decoder head. Table 4 shows these linear probing results for all GFMs. Among the GFMs, Clay and Galileo exhibit notably stronger performance. Both models use smaller token patch sizes (8×8 for Clay and 4×4 for Galileo), compared to most of the other evaluated models that operate at a patch size of 16×16. These finer patch resolutions produce higher-granularity spatial features that align more naturally with field-level segmentation. In addition, both models robustly handle missing spectral channels: Clay is explicitly trained with missing-band augmentation, and Galileo incorporates consistent band masking and normalization statistics for partially observed inputs.

Consistency varies by geography and overlap window size. Figure 6 illustrates how consistency changes with the size of the overlapping region between the four corner crops used to measure translation sensitivity. When the overlap window is large, approaching the patch size, consistency primarily reflects intrinsic translation sensitivity, which captures the model's architectural tendency to produce different outputs for identical content at different spatial positions. When the overlap window is small, consistency also reflects context dependency, indicating how much the model's predictions vary based on surrounding image content.

PRUE achieved two to three times higher consistency than the FTW baseline across all overlap window sizes, demonstrating improvements in both intrinsic translation robustness, an architectural property, and reduced context dependency, a learned behavior. Consistency varied substantially across countries and correlates with test set difficulty. For example, our best model shows $> 40\%$ pixel disagreement in Kenya at 224-pixel context shifts, compared to $< 20\%$ in Switzerland. This suggests that consistency metrics may serve as an out-of-distribution detection signal, where low consistency during inference could indicate that the model is operating on data that differs from its training distribution, and may thus require human review or model retraining.

Consistency as a reliability signal. To quantify the relationship between consistency and performance, we computed the average consistency over test samples per country and examined its correlation with object F1 (See Figure7). We observe that consistency is weakly correlated with performance: the FTW baseline yields R^2=0.48 and PRUE-EF7 yields R^2=0.30. This indicates that consistency metrics explain *some* drops in out-of-distribution (OOD) performance but not all—a model can have high consistency but poor performance. Consistency metrics may serve as one signal among several for OOD detection, but should not be used as a sole indicator of model reliability. In deployment, low consistency scores are most useful for flagging regions that exhibit gridding artifacts and spatial prediction instability, warranting human review.

C. Per-Country Evaluation

We report full per-country metrics for all FTW countries with complete labels, excluding presence-only regions. For each country, we provide pixel IoU and object-level F1.

To represent the major model families, we evaluated the strongest-performing model from each category, as identified in Table 1 of the main paper: the FTW baseline for semantic segmentation [34], Mask2Former (M2F) with a Swin Small backbone for panoptic segmentation [8], Clay with our convolutional decoder for geospatial foundation models [9], and PRUE. All metrics were computed on each country's official FTW test split, following the evaluation protocol described in §3.4.

Across regions, the results show consistent patterns. As shown in Table 6, PRUE consistently achieved the highest or second-highest Pixel IoU and Object F1 across nearly all tested FTW countries, highlighting its robust and reliable performance across diverse geographies and agricultural systems.

D. Mosaicking and Large Scale Inference

Deploying PRUE at the country scale requires constructing spatially complete, cloud-free Sentinel-2 composites from

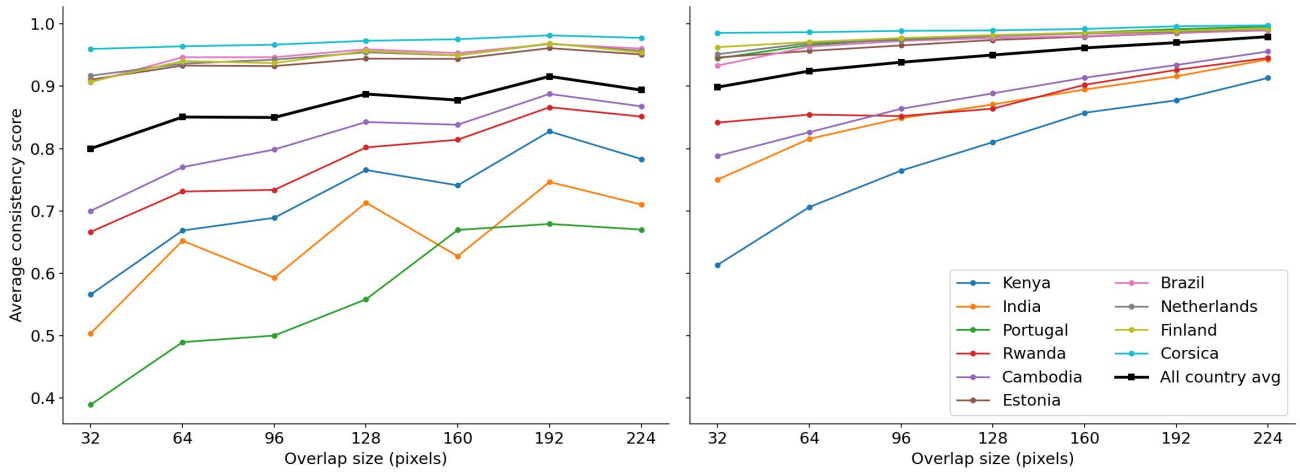

Figure 6. **Consistency dependence on overlap size.** Consistency scores as a function of overlap window size for the FTW baseline (left) and PRUE (right). Larger overlaps measure intrinsic translation sensitivity, while smaller overlaps additionally capture context dependency. Shown: top 5 and bottom 5 countries by mean consistency for each model. PRUE achieves higher consistency across all countries and overlap sizes, with particularly strong improvements in challenging regions. Hard countries (Kenya, India) show low consistency even at large overlaps, while well-represented regions (Finland, Netherlands) have high consistency, suggesting consistency metrics can identify out-of-distribution samples at inference time.

Table 6. Per-country performance comparison for the top-performing models of each architecture family: FTW baseline (semantic), Mask2Former with Swin-S (instance/panoptic), Clay (frozen GFM), Clay-FT (finetuned GFM), and PRUE (our optimized semantic model). Bold indicates best performance per country. PRUE achieves the highest or second-highest pixel IoU scores across nearly all countries. All models evaluated using the protocol in Section 3.4 on presence/absence labeled countries only.

Country	**Pixel IoU**					**Object F1**				
	FTW	M2F	Clay	Clay-FT	PRUE	FTW	M2F	Clay	Clay-FT	PRUE
Austria	0.71	0.74	0.69	0.76	**0.78**	0.41	0.38	0.39	0.47	**0.50**
Belgium	0.75	0.78	0.73	0.79	**0.82**	0.57	0.57	0.57	0.62	**0.66**
Cambodia	0.40	0.19	0.27	0.40	**0.66**	0.19	0.10	0.09	0.19	**0.36**
Corsica	0.45	**0.52**	0.47	0.51	0.51	0.18	0.24	0.18	0.22	**0.24**
Croatia	0.67	0.70	0.64	0.71	**0.77**	0.28	0.34	0.25	0.33	**0.45**
Denmark	0.83	0.57	0.83	**0.86**	**0.86**	0.52	0.24	0.51	0.58	**0.65**
Estonia	0.80	0.82	0.80	0.83	**0.84**	0.44	**0.54**	0.43	0.48	**0.54**
Finland	0.83	0.85	0.81	0.85	**0.87**	0.55	0.54	0.53	0.59	**0.64**
France	0.79	0.80	0.78	0.81	**0.83**	0.55	0.55	0.54	0.58	**0.63**
Germany	0.79	0.77	0.78	**0.80**	0.79	0.41	0.44	0.39	0.42	**0.47**
Latvia	0.81	0.84	0.81	0.84	**0.85**	0.44	0.54	0.44	0.49	**0.56**
Lithuania	0.74	0.78	0.74	0.78	**0.79**	0.39	0.48	0.38	0.45	**0.50**
Luxembourg	0.79	0.80	0.76	0.82	**0.85**	0.49	0.37	0.46	0.53	**0.56**
Netherlands	0.75	0.78	0.74	**0.81**	**0.81**	0.48	0.51	0.48	0.54	**0.57**
Portugal	0.12	0.21	0.23	**0.37**	0.10	0.03	**0.08**	0.04	0.07	0.03
Slovakia	0.92	0.91	0.92	**0.94**	**0.94**	0.53	0.61	0.53	0.58	**0.65**
Slovenia	0.58	0.66	0.55	0.65	**0.68**	0.24	0.28	0.20	0.27	**0.33**
South Africa	0.80	0.80	0.78	0.81	**0.82**	0.53	**0.56**	0.50	**0.56**	0.54
Spain	0.73	0.70	0.69	0.75	**0.83**	0.24	0.26	0.21	0.26	**0.33**
Sweden	0.81	0.82	0.80	0.84	**0.85**	0.45	0.51	0.44	0.50	**0.55**
Vietnam	0.46	0.30	0.31	0.46	**0.67**	0.15	0.09	0.08	0.15	**0.22**

irregularly sampled, partially cloudy observations. This section details our operational pipeline for scene selection, temporal compositing, and imagery quality mosaicking using latitude-based season heuristics, greedy scene selection to minimize redundancy, and cloud-optimized data formats enabling scalable parallel inference.

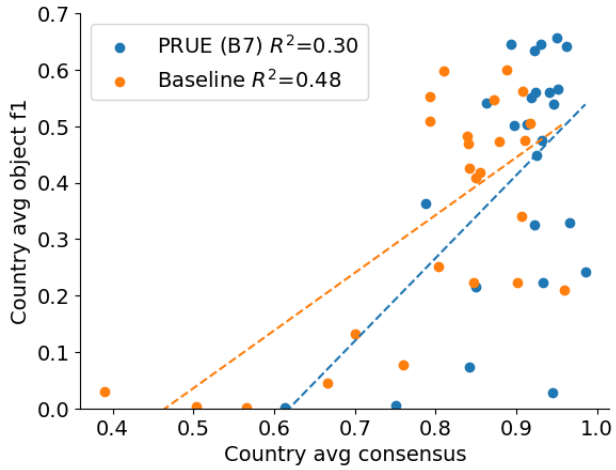

Figure 7. **Country-level consistency vs. object F1.** Each point represents one FTW country. Dashed lines show linear fits for the FTW baseline (orange, R²=0.48) and PRUE-EF7 (blue, R²=0.30). Consistency is weakly correlated with performance, suggesting it may serve as a partial out-of-distribution signal but not a reliable sole predictor of model accuracy.

Latitude-Based Season Heuristics. Planting and harvest windows were estimated using latitude-dependent day-of-year (DOY) ranges that account for hemispheric differences and climatic zones. This heuristic approach provides reasonable temporal constraints for scene selection, although we acknowledge that date selection could be substantially improved by integrating additional information on geographical variation in crop growth cycles, such as crop calendars, phenological models, or ground-based information on local planting and harvest periods.

1. Planting Season Heuristic Algorithm

function PLANTINGSEASONDOY(latitude)
  abs_lat $\leftarrow$ |latitude|
  if abs_lat > 45 **then** *High latitudes*
    return $(91, 151)$ if latitude > 0 else $(274, 334)$
  else if $20 <$ abs_lat ≤ 45 **then** *Mid-latitudes*
    return $(60, 120)$ if latitude > 0 else $(244, 334)$
  else if $5 <$ abs_lat ≤ 20 **then** *Subtropics*
    return $(121, 212)$ if latitude > 0 else $(305, 365)$
  else *Equatorial* $|lat| \leq 5$
    return $(60, 121)$
  end if
end function

2. Harvest Season Heuristic Algorithm

function HARVESTSEASONDOY(latitude)
  abs_lat $\leftarrow$ |latitude|
  if abs_lat > 45 **then** *High latitudes*
    return $(244, 304)$ if latitude > 0 else $(60, 151)$
  else if $20 <$ abs_lat ≤ 45 **then** *Mid-latitudes*
    return $(213, 304)$ if latitude > 0 else $(32, 120)$
  else if $5 <$ abs_lat ≤ 20 **then** *Subtropics*
    return $(274, 365)$ if latitude > 0 else $(91, 181)$
  else *Equatorial* $|lat| \leq 5$
    return $(182, 243)$
  end if
end function

Feature Selection via Greedy Search Optimal scene selection maximizes spatial coverage while minimizing redundancy. Input scenes were pre-filtered by cloud cover ($< 75\%$) and Scene Classification Layer (SCL) quality flags (excluding classes 1, 3, 7, 8, 9, 10, and nodata=0). The greedy approach prioritizes scenes that contribute valid observations to underrepresented spatial regions within a tile, enabling the use of relaxed scene-level cloud cover thresholds. Scenes with high overall cloud cover may still contain substantial cloud-free areas that fill critical gaps in the composite, thereby improving spatial completeness without requiring additional acquisitions.

Cloud-Optimized GeoTIFF Storage. For each Sentinel-2 grid tile and temporal period, median composites were constructed from the selected scenes. Spectral bands (B02, B03, B04, B08) at native 10 m resolution were masked using SCL upsampled from 20 m via nearest-neighbor. Temporal medians were computed alongside valid observation counts. Outputs were stored as float32 Cloud-Optimized GeoTIFFs with 1024×1024 internal tiling.

GTI-Based Reprojection, Resampling and Zarr Assembly. GDAL Tile Index (GTI)[1] files provide virtual mosaics referencing distributed COGs. During Zarr[2] construction,

[1] https://gdal.org/en/latest/drivers/raster/gti.html
[2] https://zarr.readthedocs.io/en/stable/

3. Greedy Scene Selection Algorithm

function SELECTSCENESGREEDY(valid_mask, target_coverage $= 5$, max_scenes $= 10$)
  Input: valid_mask - boolean array of shape (T, H, W)
  coverage_depth $\leftarrow \mathbf{0}_{H \times W}$
  remaining $\leftarrow \{0, 1, \ldots, T-1\}$
  selected $\leftarrow$ []
  for $i = 1 \rightarrow$ max_scenes **do**
    best_idx $\leftarrow$ NULL
    best_gain $\leftarrow -1$
    for each idx $\in$ remaining **do**
      undercovered $\leftarrow$ (coverage_depth $<$ target_coverage)
      new_valid $\leftarrow$ valid_mask[idx]$\wedge$undercovered
      gain $\leftarrow \sum$ new_valid
      if gain $>$ best_gain **then**
        best_gain $\leftarrow$ gain
        best_idx $\leftarrow$ idx
      end if
    end for
    if best_gain $= 0$ **then**
      break
    end if
    selected.APPEND(time[best_idx])
    coverage_depth $\leftarrow$ coverage_depth $+$ valid_mask[best_idx]
    remaining.REMOVE(best_idx)
  end for
  return selected
end function

`gdal_translate` performs windowed extraction with on-the-fly reprojection to EPSG:3857 (Web Mercator) using nearest neighbor resampling and writes to a temporary local file. Reprojection to EPSG:3857 prior to inference eliminates the need for downstream pipelines to perform coordinate transformations, and enables global non-overlapping results without tile artifacts. The windowed data is then loaded and inserted directly into the Zarr store, with spatial partitions written to Zarr v3 arrays in parallel with Ray[3]. This creates a robust and scalable approach to building large-scale, cloud-optimized data ready for downstream analysis.

E. Large Scale Inference Visual Samples

To qualitatively assess PRUE's performance at operational scale, we present in Figure 8 representative visual samples from each country-scale deployment described in Section 5. These samples illustrate model behavior across diverse agricultural systems (Japan, Mexico, Rwanda, South Africa, Switzerland) spanning a wide range of field sizes. The visualizations demonstrate PRUE's ability to maintain spatial consistency across large extents under atmospheric variation. Visual inspection reveals spatial patterns of success and failure modes that inform future improvements.

[3] https://docs.ray.io/en/latest/index.html

F. Change Detection Analysis

The change detection visualizations presented in Figure 8 are intended to demonstrate how multi-year maps produced by PRUE can signal probable field-scale changes. We leave more detailed studies of change detection to future work.

The method computes the absolute difference between semantic logits from consecutive years, applies min–max normalization, and thresholds at 0.5 to obtain a binary change mask. For well-calibrated models, this threshold highlights high-confidence semantic shifts. Visual inspection confirms that even small-scale detected changes are consistent with cultivation shifts (e.g., fields appearing or disappearing between years), and that artifacts from misregistration and atmospheric variation are uncommon. We note that lacking ground truth change labels, we relied on photo-interpretation of high-resolution basemap imagery rather than quantitative accuracy assessments, which we leave as future work.

G. Future Directions

Several directions remain open for future work, including: (1) comprehensive object-level [46] and thematic accuracy [43] assessments on country-scale deployments with independent reference data; (2) exploring deployment metrics as out-of-distribution detectors (our preliminary findings suggest low consistency scores may signal when models encounter dissimilar data); (3) incorporating super-resolution approaches for delineating smallholder fields from temporal stacks of imagery; and (4) systematic post-processing ablations (morphological operations, topological cleaning, confidence thresholding) to further improve boundary quality in challenging regions.

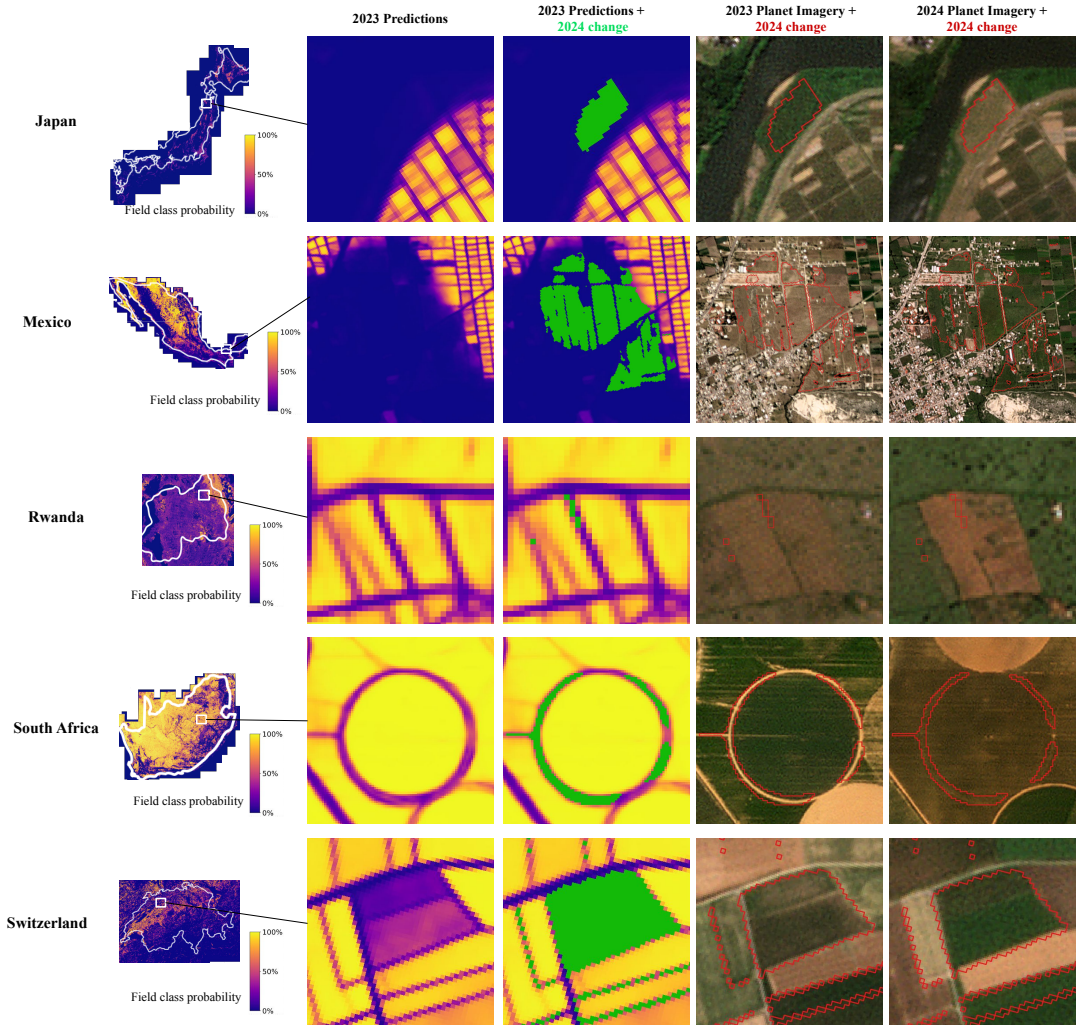

Figure 8. Example visuals over each country in our large-scale inference set (Japan, Mexico, Rwanda, South Africa, and Switzerland). For each region of interest, we show: (1) the PRUE field boundary predictions from 2023; (2) the 2023 predictions with changes detected in 2024 highlighted in bright green; (3) Planet monthly basemaps from 2023 with the vectorized change mask outlined in red; and (4) Planet monthly basemaps from 2024 with the vectorized change mask outlined in red. The basemaps shown for each pair are from the same month in consecutive years.