TESSERA V2: SCALING PIXEL-WISE EARTH FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Pixel-wise Earth-observation (EO) foundation models are now achieving state-of-the-art performance via generated spatial embeddings. However, how these models scale and how best to spend a fixed pretraining budget remain poorly understood. We present the largest controlled scaling study for EO to date: 395 training runs within a fixed pixel-wise BARLOW TWINS family, each evaluated on 15 diverse downstream tasks. We find that pretraining loss barely predicts downstream performance ($|\text{Pearson } r| < 0.2$), so selecting models by loss wastes a large share of the compute. We also find that, as the training budget grows, the encoder and the data should grow together while the projector stays fixed, which gives a simple rule for allocating compute. Using this rule, we train a family of pixel-wise teachers (0.5B, 1B, and 2B) and distil the largest into compact students for embeddings-as-data deployment. In aggregate, our 44-million-parameter distilled student outperforms every open and proprietary embedding product we test, several of them an order of magnitude larger. These students produce Matryoshka representations that are inexpensive to serve: a 16-dimensional prefix keeps 92% of the full 128-dimensional performance at 1/8 of the storage. Together, these results give a concrete, empirically grounded recipe for scaling pixel-wise EO foundation models: train large encoders, select by downstream performance, and distil into flexible student models. We plan to release global 10 m annual embeddings covering 2017–2025 as version 2 of the Tessera foundation-model embeddings product.

1 INTRODUCTION

A key bottleneck in using Earth observation (EO) for downstream tasks is making the satellite data analysis-ready. This involves complex steps such as radiometric calibration, cloud and atmospheric correction, cross-sensor harmonisation, and expensive computation over raw imagery. The ground-truth labels required are often scarce and tied to specific regions and seasons (Metcalfe et al., 2025; Hou et al., 2026). Satellite observation data also intrinsically differ from standard RGB images because each satellite observes a given pixel at irregular cadences, cloud cover can obscure optical sensors (Figure 1a), and cloud-free composites remove temporal dynamics that might be highly informative for certain downstream tasks (Zeng et al., 2020; Xiao et al., 2025). A model-derived spatially-explicit representation (or *embedding*) layer for global surface state lowers these barriers.

Earth embeddings are vector representations of the observations of a specific place over some temporal context, from a single acquisition to a multi-year time series, that compress and fuse multi-source observations (Klemmer et al., 2025b; Fang et al., 2026). Provided at global scale, in the 'embeddings-as-data' approach, they free analysts from acquiring and storing heterogeneous raw data, from complex pre-processing, and from the need for expensive user-side GPU compute. When used with lightweight task-specific *heads* they have been shown to match, and often outperform, *ad hoc* models that rely on traditional remote sensing data.

An ideal embedding product should be *analysis-ready*, that is, not requiring the processing of raw imagery or running GPU inference; *transferrable* across tasks, workflows, and regions rather than tuned to one benchmark (Klemmer et al., 2025b; Lyu et al., 2026); *reproducible*, with inspectable and extensible algorithms and weights; *economical* in the resources that limit EO adoption, namely labels, compute and its carbon footprint, storage, I/O, and engineering time; and *adaptive*, exposing an explicit accuracy-versus-cost trade-off instead of locking users into a single model size and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

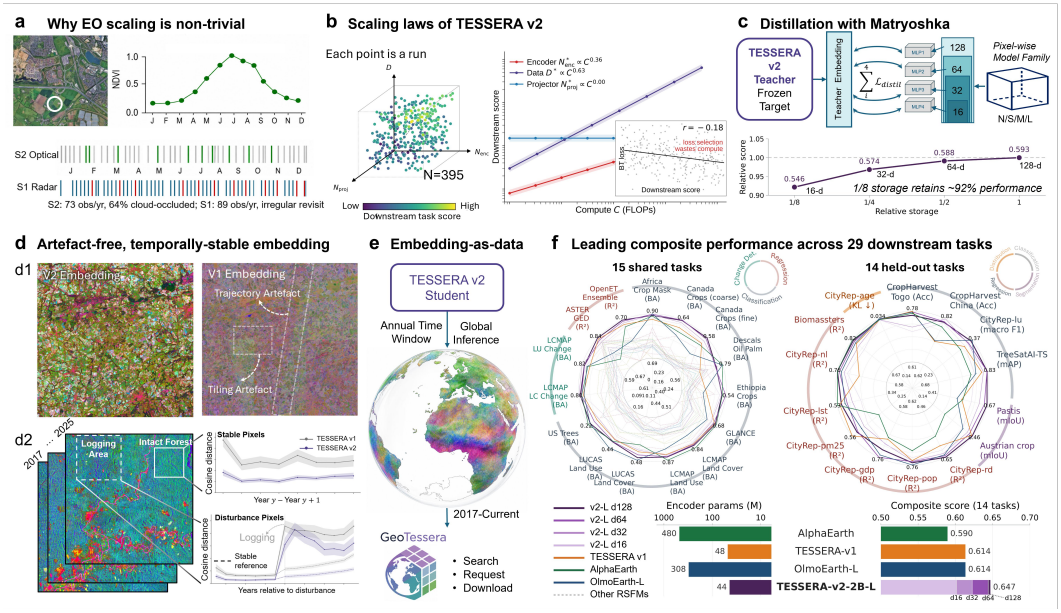

Figure 1: Overview of Tessera v2. **(a)** Sparse, irregular Sentinel-2/Sentinel-1 sampling at one location. **(b)** 395-run downstream-driven scaling sweep and compute-optimal fits; inset: pretraining loss vs. downstream score. **(c)** MATRYOSHKA distillation of the $2\,B$ teacher into N/S/M/L students, $d \in \{16, 32, 64, 128\}$. **(d)** Artefact removal and inter-annual stability vs. v1. **(e)** Deployment as an analysis-ready data product.**(f)** Leading composite performance: radars give per-task scores on the shared ALPHAEARTH suite (left) and the held-out datasets (right), bars the held-out composite (Tessera v2-2B-L, shades $d \in \{16, 32, 64, 128\}$) and encoder size.

embedding dimensionality. Existing systems each satisfy only a subset of these requirements, as discussed next.

Remote sensing foundation models (Cong et al., 2022; Reed et al., 2023; Fuller et al., 2023; Guo et al., 2024; Tseng et al., 2024; 2025; Astruc et al., 2025) learn strong representations but rarely release data products, so users face pre-processing and inference costs (Fang et al., 2026). Specialised archives are useful within a limited scope—for example, ESD (Chen et al., 2026) compresses 25 years of Landsat/MODIS reflectance into quantised 30 m embeddings—but do not provide 10 m resolution or SAR. ALPHAEARTH (Brown et al., 2025) is analysis-ready at global scale but, although the embeddings are public, the training procedure and weights are not (Hou et al., 2026). Tessera v1 (Feng et al., 2026) comes closest to meeting these requirements, being analysis-ready, label-efficient, open, and reproducible, but it produces a single fixed 128-dimensional embedding, so every user is faced with the same storage and I/O cost regardless of their deployment constraints. To our knowledge, no existing system is an open, analysis-ready *family* in which users pick the model size and embedding dimension that fits their budget (see Appendix A for details). This is the research gap we address in this work.

Closing this gap requires changes to the model training pipeline. When training language and vision models, *scaling laws* have been empirically derived to decide how a pretraining budget should be allocated to training each part of a model based on the pretraining loss (Kaplan et al., 2020; Hoffmann et al., 2022; Zhai et al., 2022). In EO, using such a loss is sub-optimal, because cloud cover and orbital sampling dominate input variance and a redundancy-reduction objective can be minimised through model invariances that might carry no downstream value. This requires scaling laws to be based on performance across a suite of downstream tasks, which few existing EO scaling studies do (Dionelis et al., 2025; Wickrema et al., 2025). We therefore ran a downstream-driven scaling study, with 395 controlled pretraining runs of pixel-wise BARLOW TWINS (Zbontar et al., 2021; Lisaius et al., 2024) encoders (the backbone family of Tessera v1),[1] each evaluated on 15 diverse downstream tasks. As expected, pretraining loss proved a weak proxy for downstream utility and

[1]The sweep ran on 1,024 NVIDIA GH200 superchips, each pairing a Grace CPU with one H100 GPU (96 GB HBM3).

selecting models using this approach needs roughly 2–5× the compute to match downstream-driven selection (Finding **F1**). Our downstream-optimal allocation instead increases encoder capacity and training data together while the projector stays essentially fixed (Finding **F2**).

Scaling the encoder in this way increases the inference cost, which dominates any embeddings-as-data product. We therefore recover deployment efficiency through *distillation* (Hinton et al., 2015). Specifically, we train a teacher with a large encoder then distil it into a family of compact students. This allows adaptivity along two axes: a choice of student sizes lets users who generate their own embeddings trade compute for quality. Additionally, MATRYOSHKA prefixes expose 16-, 32-, 64-, and 128-dimensional views of embeddings for users who consume embeddings as data.

We find that using a large encoder allows Tessera v2 to achieve state-of-the-art performance across a diverse set of tasks. Across a 29-task suite (the 15-task ALPHAEARTH suite plus 14 further held-out datasets), the distilled students outperform every open and proprietary embedding product we compare, and even the smallest prefixes keep most of the full-dimensional score. Interestingly, the embeddings are also largely free from the banding artefacts visible in v1 (Figure 1d).

In summary, our contributions are:

1. **Downstream task-driven empirical scaling laws.** Across 395 runs evaluated on 15 downstream tasks, we find that the pretraining loss is a poor selection target (**F1**), and that the compute-optimal allocation grows encoder size and data volume together while the projector stays fixed (**F2**).
2. **A pixel-wise embedding product family: Tessera v2.** Guided by this rule, we train large teachers and distil them into compact students that lose at most 0.015 of composite score at a deployment cost 50× to over 300× below the teacher's.
3. **Storage-adaptive MATRYOSHKA embeddings.** Each student exposes Matryoshka prefixes at $d \in \{16, 32, 64, 128\}$ from one embedding, providing an accuracy/storage knob without requiring retraining. The d=16 embedding has ∼92% of the d=128 score at only 1/8 of the storage.

2 RELATED WORK

The present work builds on Tessera v1 (Feng et al., 2026), which uses BARLOW TWINS redundancy reduction to cloud-corrupted EO time series following Lisaius et al. (2024). We reuse its d-pixel formulation and pretraining recipe as the fixed model family for the scaling study.

EO foundation models. Most remote-sensing foundation models pretrain *spatial* backbones on single-time and often cloud-filtered patches with contrastive or masked-image objectives (Mañas et al., 2021; Cong et al., 2022; Reed et al., 2023; Guo et al., 2024), with multi-sensor and multi-resolution variants fusing optical and SAR or ingesting many sensors at once (Fuller et al., 2023; Xiong et al., 2024; Astruc et al., 2025; Tseng et al., 2025), and hierarchical spatiotemporal transformers for satellite image time series (Qin et al., 2025). PRESTO (Tseng et al., 2024) and EARTHPT (Smith et al., 2023) instead process per-pixel time series (EARTHPT as an autoregressive forecaster), and MOSAIKS (Rolf et al., 2021) explored lightweight universal features (see Appendix A for details). These designs require per-task fine-tuning of the backbone, which is compute- and label-intensive. Moreover, they have not carried out a scaling law analysis to determine how to optimally allocate limited compute resources, as we have, in Section 3.

Embedding products. A second line of research focuses on providing analysis-ready embeddings instead of a backbone. Tessera v1 releases global annual 10 m pixel-wise int8 embeddings along with an accompanying retrieval library (Feng et al., 2026), ALPHAEARTH provides global annual 10 m embedding fields by fusing data from many sensor (Brown et al., 2025), ESD (Chen et al., 2026) compresses 25 years of Landsat/MODIS reflectance into quantized 30 m embeddings, and OLMOEARTH (Herzog et al., 2026) computes embeddings on demand, allowing custom time windows at the cost of per-request inference. Their shared limitation is that users cannot obtain less-costly or lower-dimensional embeddings if they are willing to sacrifice task performance. Tessera v2 adds two degrees of freedom. First, it generates students at four sizes, produced by scaling-law-guided distillation. Second, it provides MATRYOSHKA prefixes $d \in \{16, 32, 64, 128\}$ from a single embedding without retraining. We note that evaluating such embedding families has begun to attract dedicated benchmarks (Vinge et al., 2025).

Scaling laws. Empirical scaling laws have shaped recent language and vision work (Kaplan et al., 2020; Hoffmann et al., 2022; Zhai et al., 2022), and are typically fit to a self-supervised pretraining loss on the assumption that the loss proxies downstream quality. This assumption can fail even in language modelling, where a lower pretraining loss does not reliably translate into better downstream performance (Hu et al., 2025). The two EO studies of scaling laws that we are aware of do not test the role of pretraining loss directly. Dionelis et al. (2025) sweep architecture, size, and data on PhilEO Bench (Fibaek et al., 2024) but simultaneously vary both architecture and capacity over a coarse grid, entangling the two. Wickrema et al. (2025) fit peta-pixel power laws to the validation loss in a data-limited regime that, as they note, is confounded by under-trained large models, and defer the loss-vs-downstream comparison to future work. Neither isolates encoder, projector, and data contributions using a fixed compute budget. In contrast, we isolate all three and fit the compute allocation against task performance directly, evaluating every run on 15 downstream tasks.

Distillation and nested embeddings. It is widely known that knowledge distillation (Hinton et al., 2015) transfers representations from a high-capacity teacher to a compact student. Similarly, MATRYOSHKA representation learning (Kusupati et al., 2022) produces nested embeddings whose prefixes work at multiple dimensionalities. Yet, the two are rarely combined in EO, and rarely with attention to whether the self-supervised objective can support nested coordinates. We find empirically that adding prefix losses to pretraining does not (Appendix W.2), whereas distillation against a *fixed* teacher does; we discuss a possible reason in Section 6.

3 DOWNSTREAM-DRIVEN SCALING LAWS

This section answers the question: within a fixed architectural family and a given compute budget, how should the budget be split between encoder size, projector size, and training data volume? The experiments below describe the family we sweep, pixel-wise Sentinel-1/2 encoders pre-trained with BARLOW TWINS (Zbontar et al., 2021). Note that we do **not** view these as 'universal' EO scaling laws (please refer to the discussion in Appendix S.1).

3.1 STUDY DESIGN

Architecture choice. Attributing performance differences to projector or encoder size or data is only meaningful when the architecture is held constant. We therefore select it first, sweeping ten structural axes one at a time by aggregated downstream score (Appendix F), and hold the selected configuration fixed for the entire scaling sweep.

Controlled sweep. With architecture fixed, we pretrain 395 models on 1,024 H100 GPUs in an iso-FLOP grid for encoder size N_{enc} (16 widths, 7–278 M), projector size N_{proj} (four widths, $P \in \{2048, \ldots, 16384\}$, i.e. 21 M–1.3 B), and training data D (0.03–3,202 M d-pixels). At each compute level, the compute-optimal size is computed as the vertex of a quadratic equation fit to $\log N$ (details in Appendix G). The compute axis is the total training FLOPs,

$$C \;=\; 18\,D\left(N_{\text{enc}}\,\bar{L} + N_{\text{fus}}\,M + N_{\text{red}} + N_{\text{proj}}\right) \;+\; 6\,D\,P^2, \qquad \bar{L} = 12, \tag{1}$$

where $18 = 6 \times 3$ is the $6ND$ factor times the *three* encoder passes per step (the two BARLOW TWINS augmentation views and the mixup view, the fourth shuffled view being a permutation that needs no pass (Bandara et al., 2023)); $\bar{L} = 12$ is the mean per-view token length actually processed (the augmentation samples $L \in \{8, 16\}$ each step); N_{fus} and N_{red} are the modality-fusion and dimensionality-reduction costs, $N_{\text{proj}} = 5P^2$ the projector, and $6DP^2$ the BARLOW TWINS cross-correlation cost. The per-token derivation, the omitted $\mathcal{O}(DL^2)$ self-attention term ($<1\%$ of C), and a `FlopCounterMode` (PyTorch Team, 2024) check are in Appendix B.

Downstream evaluation. Every pretrained model is also evaluated on 15 ALPHAEARTH suite tasks drawn from 10 source datasets (classification, change detection, regression), with chance-adjusted metrics averaged into one composite downstream score, the y-axis of Figure 2 (Appendices L and O). On the same suite Tessera v1 scores 0.541 and ALPHAEARTH 0.560. Both appear as baselines in Figure 2a–c. Since these are full-budget production systems while the grid deliberately spans small, data-limited configurations to trace the compute frontier, it is the upper envelope of the sweep, not

any individual run, that approaches them. For every run, we also record the converged BARLOW TWINS loss on held-out d-pixels, normalised across projector widths (Appendix S.3).

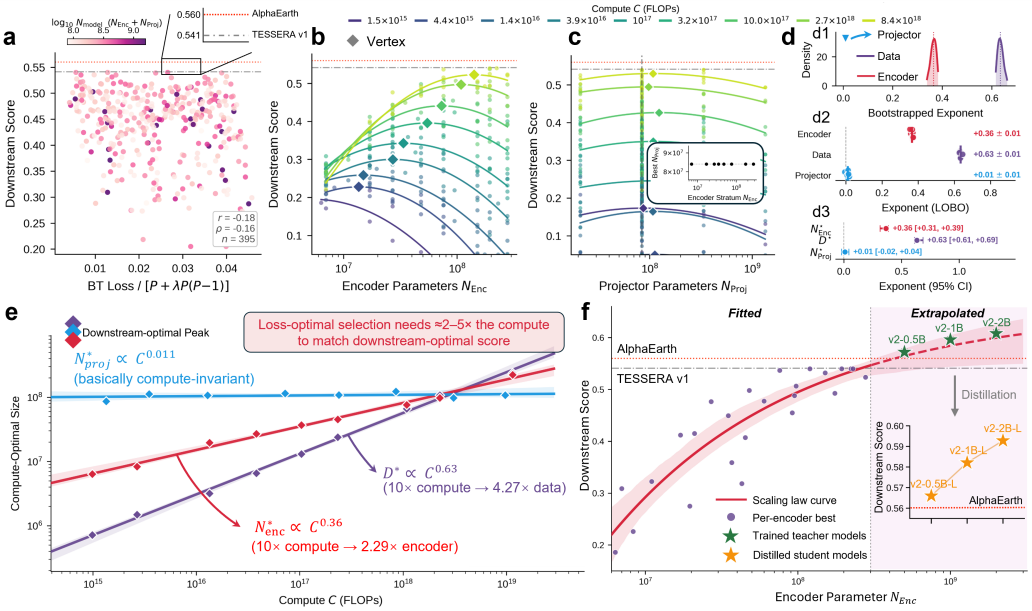

Figure 2: Downstream-driven scaling laws (395 runs; composite score over 15 tasks). **(a)** Pretraining loss vs. downstream score, one point per run, coloured by model size ($r = -0.18$). **(b, c)** Iso-FLOP parabolas for encoder and projector; diamonds mark the compute-optimal size (parabola vertex) per compute bucket. Inset in (c): the best N_{proj} is flat across encoder strata. **(d1–d3)** Bootstrap density, leave-one-bucket-out, and 95% CIs of the three exponents. **(e)** Power-law fits to the vertices, $N^{\star}_{\text{enc}} \propto C^{0.36}$, $D^{\star} \propto C^{0.63}$, $N^{\star}_{\text{proj}} \propto C^{0.011}$. **(f)** Out-of-sample validation: the encoder law fit on the sweep (solid, $\leq$278 M) extrapolates (dashed band) to the 0.5/1/2 B teachers (stars), none used in the fit. Inset: distilled students.

3.2 FINDING 1: THE PRETRAINING LOSS IS NOT A GOOD PREDICTOR OF DOWNSTREAM TASK PERFORMANCE

Across the 395 runs, the converged BARLOW TWINS loss and the composite downstream score are nearly independent (Figure 2a; Pearson $r = -0.18$, Spearman $\rho = -0.16$); Appendix S.2 discusses why. Moreover, we find, by fitting separate power laws through loss-selected and score-selected bucket peaks (Appendix S.3), that loss-based selection needs roughly 2–5$\times$ the compute to reach the same downstream score. EO scaling laws for this family must therefore be fit against downstream metrics rather than pretraining loss.

3.3 FINDING 2: ENCODER SIZE AND DATA REQUIREMENTS SCALE WITH COMPUTE; THE PROJECTOR DOES NOT

As before, we group runs into nine iso-FLOP buckets and within each bucket fit a quadratic equation to model $\log N$. The vertex of the quadratic is thus the compute-optimal size at that budget. We find that encoder vertices shift right as compute grows (Figure 2b), while projector vertices stack along a vertical line (Figure 2c). Power-law fits through the vertices (Figure 2e) give $N^{\star}_{\text{enc}} \propto C^{0.36}$ (95% CI $[+0.31, +0.39]$), $D^{\star} \propto C^{0.63}$ ($[+0.61, +0.69]$), and $N^{\star}_{\text{proj}} \propto C^{0.011}$ ($[-0.02, +0.04]$). Full fit statistics are in Appendix S.4. The encoder and data exponents sum to ≈ 1 ($0.36 + 0.63 = 0.99$): on the iso-FLOP vertices $C \propto N_{\text{enc}} D$ pins $D^{\star}$ once $N^{\star}_{\text{enc}}$ is set (Appendix G), so the informative quantity is how compute *splits* between encoder and data, not the sum. Since the projector exponent is statistically indistinguishable from zero, the compute-optimal choice is to use a relatively small projector. In contrast, increasing the encoder size improves performance.

Deriving a recipe for scaling. Since encoder capacity and data cost compute while the projector is compute-invariant, the compute-optimal use of a large budget is to use an oversized encoder trained on correspondingly more data, with the projector held near its optimum as a training scaffold. Deployment feasibility, in terms of model size and embedding dimension, is achieved through distillation (Sections 4 and 5).

The fitted law extrapolates well. We test the validity of the scaling law out of sample in Figure 2f. The encoder law, fit on the sweep alone (encoders up to 278 M), extrapolates well to the 0.5, 1, and 2 B teachers of Section 4. All three are on the predicted curve and improve monotonically with scale, from 0.572 to 0.608, performing better than both Tessera v1 (0.541) and AlphaEarth (0.560). Distilling each into its deployable student costs at most 0.015 of composite score, and the students also improve with scale (Figure 2f, inset).

4 A PIXEL-WISE TEMPORAL TEACHER

Figure 3 summarises the design of the single large encoder. Compared to Tessera v1 (Feng et al., 2026), Tessera v2 adds multi-scale temporal pretraining, adaptive full-observation inference, a unified all-Transformer architecture with cross-modal fusion, and scaling-law-guided distillation into Matryoshka students (Section 5). We describe these next.

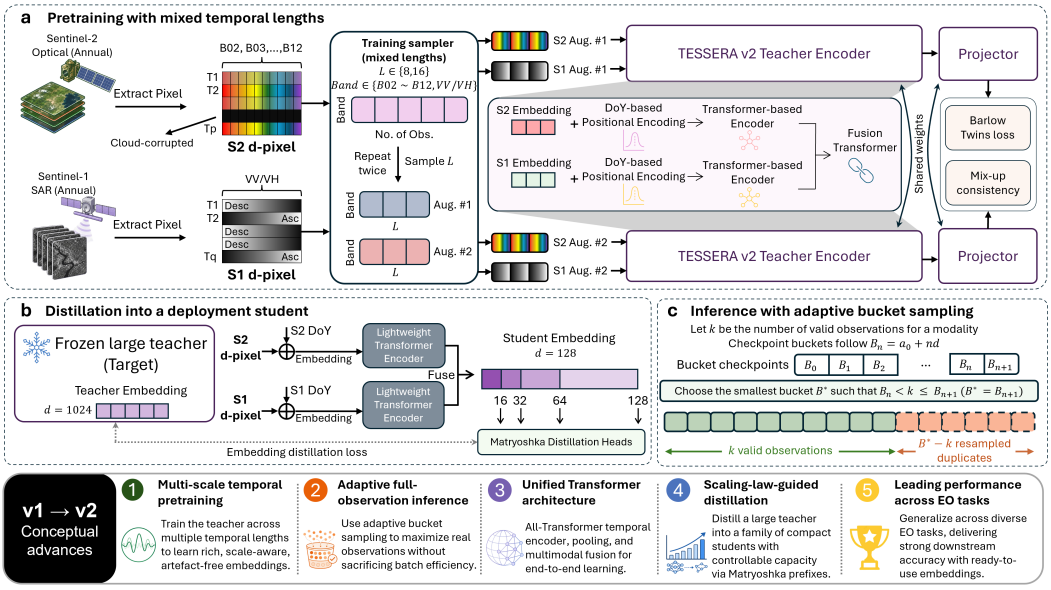

Figure 3: Tessera v2 architecture. **(a)** Pretraining: two views per d-pixel at random length $L \in \{8, 16\}$, per-modality Transformers with day-of-year encoding, cross-modal fusion, and Barlow Twins + mix-up. **(b)** Distillation: Matryoshka prefix heads at $d \in \{16, 32, 64, 128\}$ reconstruct the frozen teacher embedding. **(c)** Inference: each pixel's k valid observations are packed into the smallest bucket $B^\star \geq k$, with residual slots filled by midpoint resampling.

Inputs. A d-pixel at location (i, j) is the time series of all Sentinel-1 and Sentinel-2 observations at that 10 m pixel over one year. A binary mask marks valid timesteps (Feng et al., 2026), so every clear observation is retained, and cloud occlusion and irregular revisits are tolerated rather than imputed.

Architecture. The teacher is a 2 B dual-branch pixel-wise encoder (Figure 3a). Each modality branch embeds its valid observations, adds a sinusoidal day-of-year positional encoding, runs a per-modality Transformer, and aggregates the variable-length sequence by learned attention pooling. A fusion Transformer then combines the two modality tokens into one embedding $\boldsymbol{t} \in \mathbb{R}^{d_T}$, $d_T = 1024$. d_T deliberately exceeds the student's 128, leaving room for compression. The encoder is larger than any point in the Section 3 sweep (which tops out at 278 M). Following the recipe, we extrapolate

its $C^{0.36}$ encoder law to the $2\,B$ teacher, whose training compute is measured on the same axis (Equation (1)). Layer, head, and projector details are in Appendix H.

Objective and training. Similar to Tessera v1, we train using BARLOW TWINS over two temporally subsampled views, with an additional mix-up consistency regulariser (Bandara et al., 2023) and global shuffling of d-pixels across tiles. Unlike v1's single fixed sample length, at every step the view length is a random $L \sim \mathrm{Uniform}\{8, 16\}$, so each view is a sparse random subsample of the year that forces the encoder to recover annual phenology from only a few observations. At inference we pass *all* valid observations (Section 4). Because they are placed at their true day-of-year and pooled independent of their length, they provide more evidence about the same phenology rather than potentially out-of-distribution input. Whole modalities are also dropped with a small probability, which doubles as the training signal for a pixel with no valid observations in one modality.

The teacher is pretrained for a single epoch over $\sim$14 billion d-pixels. Loss equations, the mix-up derivation, the geographic distribution of the source patches, and systems details can be found in Appendices C, I and T.

The teacher as a distillation target. Most users cannot run a $2\,B$ pixel-wise encoder over global Sentinel-1/2 even once, let alone recurrently. The scaling laws let us determine *where* to spend pretraining compute, but what users can afford to *serve* is determined by the huge cost of global inference. We therefore treat the frozen teacher $\mathcal{T}_\theta : P_{i,j} \mapsto \boldsymbol{t}$ as a fixed *representation distribution* and generate a family of students by distilling against it (Section 5).

Adaptive bucket sampling at inference. The number of valid observations k varies from a handful in heavily clouded regions to roughly a hundred in clean ones, yet batched inference requires fixed-length inputs. Where Tessera v1 sampled a fixed $L = 40$ timesteps, discarding or duplicating observations, Tessera v2 packs each pixel into the smallest bucket from the ladder $\{16, 32, 48, \ldots\}$ that fits all k observations, filling residual slots by midpoint resampling (Figure 3c). No observation is discarded, and a batch still partitions into a few fixed-length groups that run in parallel. For details of bucket construction, please refer to Appendix D.

5 DISTILLING A DEPLOYABLE STUDENT FAMILY

We expect users to consume our *embeddings as data*, that is, as precomputed annual pixel embeddings with lightweight task heads, without running or fine-tuning a backbone (Feng et al., 2026; Brown et al., 2025). The teacher cannot support this directly because inference costs for one global, annual, 10 m Sentinel-1/2 pass grow near-linearly with the number of encoder parameters. The $2\,B$ teacher costs roughly 100 H100-years per global pass against 0.04 to 2 for the students (Appendix U). Storage and download cost, meanwhile, scale with embedding dimension, which is what the MATRYOSHKA prefixes address.

Student family. We distil the teacher into four students of the same architectural form, differing only in backbone width and depth and each emitting a 128-dimensional embedding: Tessera v2-2B-L ($44\,M$, for provider-side global inference), M ($21\,M$, a balanced default), S ($7\,M$, low-resource), and N ($1\,M$, for edge and on-device use). Distillation uses 1.2 billion d-pixels on 64 H100 GPUs (Appendices H, J and K).

MATRYOSHKA distillation objective. For each prefix length $k \in \mathcal{K} = \{16, 32, 64, 128\}$ we attach an MLP head $h_k : \mathbb{R}^k \to \mathbb{R}^{d_T}$, used only during distillation, and train the student against the frozen teacher embedding $\boldsymbol{t}$ with

$$\mathcal{L}_{\mathrm{DIST}}(\boldsymbol{s}, \boldsymbol{t}) \;=\; \sum_{k \in \mathcal{K}} \Big(1 - \cos\big(h_k(\boldsymbol{s}_{1:k}),\; \boldsymbol{t}\big)\Big). \tag{2}$$

Each prefix reconstructs the *full* teacher embedding rather than a copy of its first k coordinates, making the prefix a rate–distortion code for $\boldsymbol{t}$. At inference the heads are discarded and the user takes any prefix $\boldsymbol{s}_{1:k}$ directly. The frozen teacher's targets are cached, so compute is spent almost entirely on the student.

Nesting through distillation, not pretraining. The obvious alternative, adding a prefix-BARLOW TWINS term for each k to the pretraining loss, underperforms in our experiments; the evidence and a coordinate-exchangeability ablation are in Appendix W.2, and we offer a hypothesis for the cause in Section 6.

5.1 BENCHMARK RESULTS

We evaluate performance in two stages. First, on the 15-task ALPHAEARTH suite (classification, regression, and change detection; Appendix L) we compare 20 embedding products, remote-sensing foundation models, and generic backbones, among them the two directly comparable embedding-as-data systems ALPHAEARTH (Brown et al., 2025) and Tessera v1 (Feng et al., 2026), plus OlmoEarth (Herzog et al., 2026) (the full list is in Appendix O). Second, we take the four highest-performing of these and run a *held-out* evaluation on 14 further datasets that played no part in development, covering classification, segmentation, regression, and distribution prediction (Appendix M). The two stages together form a 29-task suite. We use the ALPHAEARTH approach for scoring, using nearest-neighbour classifiers and linear probes, and a small ($<$2M-parameter) CNN on the held-out suite where spatial context matters (Appendix O). As with all foundation models, downstream performance is sensitive to the choice of task head, so our numbers may differ from published results using a different head (Appendix P).

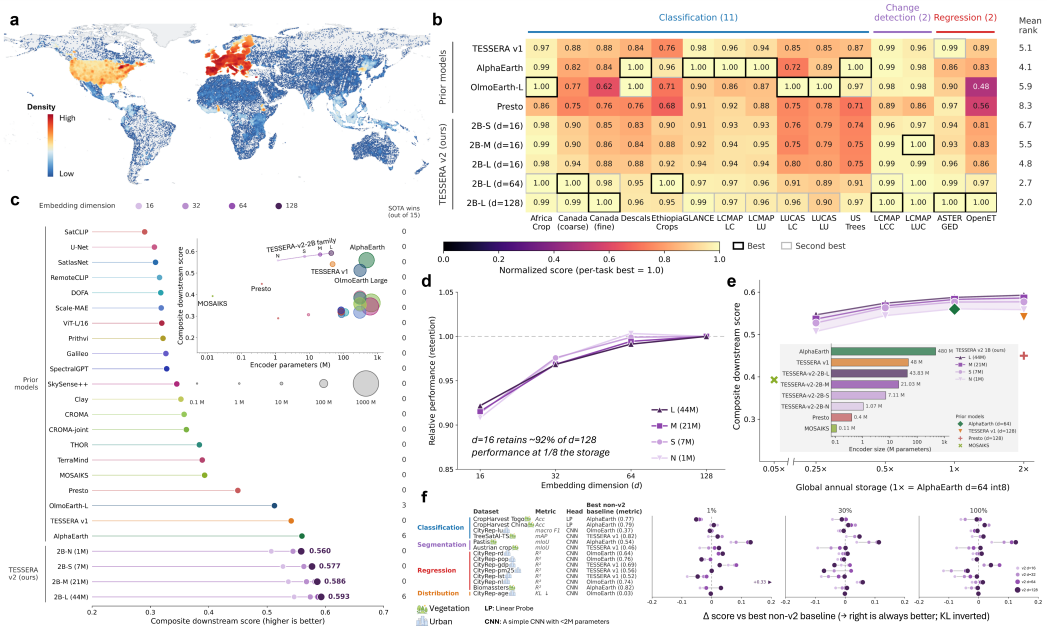

Figure 4: ALPHAEARTH suite results (15 tasks) and held-out generalisation (14 datasets). **(a)** Geographic density of all downstream labels. **(b)** Per-task heatmap with mean rank. **(c)** Composite score per model; the four markers per Tessera v2 row are $d \in \{16, 32, 64, 128\}$. **(d)** Score vs. prefix dimension. **(e)** Score vs. global annual storage, relative to ALPHAEARTH at d=64 int8. **(f)** Label efficiency at $1/30/100\%$ of the labels: Δ score vs. the best non-v2 baseline.

Headline results. The 14 held-out datasets are our primary evaluation because they played no part in developing our system, whereas the ALPHAEARTH suite labels guided v2 model and hyperparameter selection and may have contributed training data to some baselines. On the held-out suite Tessera v2-2B-L has the best composite of the four systems evaluated on it, 0.647 (seed s.d. $\sim$0.006; per-task spreads in Table 13), against OlmoEarth-L and Tessera v1 (both 0.614) and ALPHAEARTH (0.590), using a 44 M encoder an order of magnitude smaller than ALPHAEARTH (480 M) or OlmoEarth-L (308 M) (Figure 1f). On the 15 ALPHAEARTH suite tasks the composite score is 0.593 (M, S, N at 0.586, 0.577, 0.560; Figure 4c), exceeding ALPHAEARTH (0.560), Tessera v1 (0.541), and OlmoEarth-L (0.512), with the best mean rank (2.0 at d=128 vs. 4.1 and 5.1; Figure 4b). ALPHAEARTH came second on the shared tasks but last of the four on the held-out ones. We treat the

held-out suite as primary not because of this gap but because it lies outside the public label sources the baselines may have trained on, including AlphaEarth's undisclosed corpus (Appendix M).

Graceful degradation across students. Across the four students, d=16 performs at $\sim$92% of the d=128 composite at $1/8$ of the storage, with d=32 and d=64 at $\sim$97% and $\sim$99% (Figure 4d). Performance is best on land-cover and change-detection tasks and worst on fine-class-count and regression, for which d=64 is the default. Every student is on the Pareto frontier of score versus storage and encoder parameters (Figure 4e). We find that using only half the storage of AlphaEarth, Tessera v2-2B-L already exceeds it (0.574 vs. 0.560). Details of per-prefix composites are in Appendix W.5.

Held-out generalisation. The held-out datasets split into a vegetation group (tree species, crop and parcel segmentation, biomass) and an urban group (land use, road density, population, GDP, nighttime lights, pollution, surface temperature, demographics). Across $1/30/100\%$ label budgets Tessera v2 beats the best non-v2 baseline on most of them, with the widest margins in the low-label regime (Figure 4f); construction and scoring are in Appendix M.

Embedding perceptual quality and temporal stability. Relative to v1, whose embeddings show along-track striping and tile-seam discontinuities aligned with Sentinel-2/Sentinel-1 acquisition geometry, v2 has far fewer artefacts while preserving geographic structure, markedly lower consecutive-year cosine distances where land cover is stable. We also find that embeddings for a location return to a prior baseline after a temporary disturbance where v1 stays noisy (Figure 1d); the full analysis is in Appendix X.

Ablations. Coordinate-exchangeability, reconstruction, and teacher-size ablations (Appendix W) show that only distilled prefixes carry a genuine coordinate ordering, and that the student prefix is a better code for the teacher embedding than the teacher's own coordinate prefix.

6 Discussion and Conclusion

In language and vision tasks, pretraining loss tracks downstream quality closely enough to fit scaling laws directly to it. For our particular EO model training approach, this assumption fails. We found that selecting models by the loss alone needs roughly 2–5$\times$ the compute to reach the same downstream score (**F1**). This suggests that future EO scaling studies should budget for downstream evaluation rather than rely only on the pretraining loss. When selection is downstream-driven, **F2** recommends spending the budget on encoder capacity with matching data, and holding the projector size fixed. Distillation then lets one pretraining run amortise across four student sizes and four embedding dimensions. Why nested prefixes must be learned by distillation rather than by prefix losses during pretraining is not obvious. We conjecture that a redundancy-reduction objective constrains only *which* subspace the embedding spans, not how information is allocated to coordinates, since a rotated embedding achieves the same loss (the projector absorbs the rotation). Thus no coordinate ordering can emerge, whereas a fixed teacher target supplies one. Appendix V develops this conjecture further.

Limitations. Our scaling laws are empirical and apply only to pixel-wise Sentinel-1/2 encoders, one self-supervised objective, and one 15-task evaluation suite (Appendix S.1), so our budget-allocation analysis is specific to that setting; the large teacher is also expensive to pretrain. Finally, our benchmarks are mostly drawn from well-studied regions, so generalisation to under-represented climates and unseen seasons remains to be fully evaluated. Broader-impact and release considerations are in Appendix Y.

Conclusion. We present Tessera v2: a 395-run downstream-driven scaling study, a 2 B pixel-wise teacher trained by the resulting allocation rule, and a distilled Matryoshka student family that leads both the 15-task AlphaEarth suite and a 14-dataset held-out suite at 50$\times$ to over 300$\times$ lower inference cost. We hope the train-large-then-distil recipe transfers to other EO embedding products.

Reproducibility Statement

We will release training code, distilled pixel-wise student and teacher checkpoints, and a frozen embedding-as-data product.

Ethics Statement

Tessera v2 is a generic representation model for publicly available Sentinel-1/2 imagery. We discuss dual-use considerations and release practices in Appendix Y.

Use of Large Language Models

We used large language models (LLMs) as general-purpose assistance tools while preparing this paper, in three ways. First, to aid and polish writing: improving the clarity, grammar, and phrasing of author-written text. Second, for retrieval and discovery: helping surface and locate related work, which the authors then read, verified, and cited. Third, to draft sections of the paper: producing initial drafts of some passages from author-provided outlines, notes, and results, which the authors then checked, edited, and rewrote as needed. LLMs were not used for research ideation or execution, to design or run experiments, to generate or analyse results, to generate synthetic datasets, or to prove mathematical claims. All technical content, experiments, and claims are the authors' own, and the authors take full responsibility for the entire content of this paper.

References

Guillaume Astruc, Nicolas Gonthier, Clement Mallet, and Loic Landrieu. Omnisat: Self-supervised modality fusion for earth observation. In *European Conference on Computer Vision*, pp. 409–427. Springer, 2024.

Guillaume Astruc, Nicolas Gonthier, Clement Mallet, and Loic Landrieu. AnySat: One earth observation model for many resolutions, scales, and modalities, 2025. URL https://arxiv.org/abs/2412.14123.

James GC Ball, Jana Annika Wicklein, Zhengpeng Feng, Jovana Knezevic, Sadiq Jaffer, Anil Madhavapeddy, Clement Atzberger, Michele Dalponte, and David A Coomes. Geospatial foundation models enable data-efficient tree species mapping in temperate mountain forests. *Science of Remote Sensing*, pp. 100466, 2026.

Wele Gedara Chaminda Bandara, Celso M. de Melo, and Vishal M. Patel. Guarding Barlow Twins against overfitting with mixed samples. *arXiv preprint arXiv:2312.02151*, 2023.

Favyen Bastani, Piper Wolters, Ritwik Gupta, Joe Ferdinando, and Aniruddha Kembhavi. SatlasPretrain: A large-scale dataset for remote sensing image understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Christopher F. Brown, Michal R. Kazmierski, Valerie J. Pasquarella, William J. Rucklidge, Masha Samsikova, Chenhui Zhang, Evan Shelhamer, Estefania Lahera, Olivia Wiles, Simon Ilyushchenko, Noel Gorelick, Lihui Lydia Zhang, Sophia Alj, Emily Schechter, Sean Askay, Oliver Guinan, Rebecca Moore, Alexis Boukouvalas, and Pushmeet Kohli. AlphaEarth foundations: An embedding field model for accurate and efficient global mapping from sparse label data, 2025. URL https://arxiv.org/abs/2507.22291.

Shuang Chen, Jie Wang, Shuai Yuan, Jiayang Li, Yu Xia, Yuanhong Liao, Junbo Wei, Jincheng Yuan, Xiaoqing Xu, Xiaolin Zhu, Peng Zhu, Hongsheng Zhang, Yuyu Zhou, Haohuan Fu, Huabing Huang, Bin Chen, Fan Dai, and Peng Gong. Democratizing planetary-scale analysis: An ultra-lightweight Earth embedding database for accurate and flexible global land monitoring. *Earth System Science Data Discussions*, 2026. doi: 10.5194/essd-2026-57. URL https://essd.copernicus.org/preprints/essd-2026-57/. Preprint.

Clay Foundation. Clay foundation model. https://github.com/Clay-foundation/model, 2024. Accessed: 2025-02-21.

Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David B. Lobell, and Stefano Ermon. SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Mikolaj Czerkawski, Marcin Kluczek, and Jędrzej S. Bojanowski. Global and dense embeddings of earth: Major TOM floating in the latent space. *arXiv preprint arXiv:2412.05600*, 2024.

Nikolaos Dionelis, Riccardo Musto, Jente Bosmans, Simone Sarti, Giancarlo Paoletti, Peter Naylor, Valerio Marsocci, Sébastien Lefèvre, Bertrand Le Saux, and Nicolas Longépé. Scaling laws for geospatial foundation models: A case study on PhilEO bench. *arXiv preprint arXiv:2506.14765*, 2025.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Heng Fang, Adam J. Stewart, Isaac Corley, Xiao Xiang Zhu, and Hossein Azizpour. Earth embeddings as products: Taxonomy, ecosystem, and standardized access. *arXiv preprint arXiv:2601.13134*, 2026.

Zhengpeng Feng, Clement Atzberger, Sadiq Jaffer, Jovana Knezevic, Silja Sormunun, Robin Young, Madeline Lisaius, Markus Immitzer, Toby Jackson, James Ball, et al. Applications of the tessera geospatial foundation model to diverse environmental mapping tasks. *Available at SSRN 6142416*.

Zhengpeng Feng, Clement Atzberger, Sadiq Jaffer, Jovana Knezevic, Silja Sormunen, Robin Young, Madeline C Lisaius, Markus Immitzer, Toby Jackson, James Ball, et al. Tessera: Temporal embeddings of surface spectra for earth representation and analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34818–34831, 2026.

Casper Fibaek, Luke Camilleri, Andreas Luyts, Nikolaos Dionelis, and Bertrand Le Saux. PhilEO bench: Evaluating geo-spatial foundation models. In *IGARSS 2024 – IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2024.

Theodor Forgaard, Jarle H Reksten, Anders U Waldeland, Valerio Marsocci, Nicolas Longépé, Michael Kampffmeyer, and Arnt-Børre Salberg. Thor: A versatile foundation model for earth observation climate and society applications. *arXiv preprint arXiv:2601.16011*, 2026.

Alistair Francis and Mikolaj Czerkawski. Major TOM: Expandable datasets for earth observation. In *IGARSS 2024 – IEEE International Geoscience and Remote Sensing Symposium*, pp. 2935–2940. IEEE, 2024.

Anthony Fuller, Koreen Millard, and James Green. CROMA: Remote sensing representations with contrastive radar-optical masked autoencoders. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:5506–5538, 2023.

Xin Guo, Jiangwei Lao, Bo Dang, Yingying Zhang, Lei Yu, Lixiang Ru, Liheng Zhong, Ziyuan Huang, Kang Wu, Dingxiang Hu, Huimei He, Jian Wang, Jingdong Chen, Ming Yang, Yongjun Zhang, and Yansheng Li. SkySense: A multi-modal remote sensing foundation model towards universal interpretation for earth observation imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27662–27673, 2024. doi: 10.1109/CVPR52733.2024.02613.

Boran Han, Shuai Zhang, Xingjian Shi, and Markus Reichstein. Bridging remote sensors with multisensor geospatial foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27852–27862, 2024. doi: 10.1109/CVPR52733.2024.02631.

Henry Herzog, Favyen Bastani, Yawen Zhang, Gabriel Tseng, Joseph Redmon, Hadrien Sablon, Ryan Park, Jacob Morrison, Alexandra Buraczynski, Karen Farley, et al. Olmoearth: Stable latent image modeling for multimodal earth observation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34806–34817, 2026.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*, 2015.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Danfeng Hong, Bing Zhang, Xuyang Li, et al. SpectralGPT: Spectral remote sensing foundation model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Shuyang Hou, Haoyue Jiao, Ziqi Liu, Lutong Xie, Guanyu Chen, Shaowen Wu, Zhangyan Xu, Zengjie Wang, Shaoqing Tang, Yaxian Qing, Jianyuan Liang, Xuefeng Guan, and Huayi Wu. Alphaearth foundations (aef) in earth observation: A systematic review of applications and practices. *Preprints.org*, 2026. doi: 10.20944/preprints202605.0981.v1.

Nicolas Houdré, Diego Marcos, Hugo Riffaud de Turckheim, Dino Ienco, Laurent Wendling, Camille Kurtz, and Sylvain Lobry. Ramen: Resolution-adjustable multimodal encoder for earth observation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 27838–27848, 2026.

Michael Hu et al. Scaling laws are unreliable for downstream tasks: A reality check. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. arXiv:2507.00885.

Johannes Jakubik, Sujit Roy, CE Phillips, Paolo Fraccaro, Denys Godwin, Bianca Zadrozny, Daniela Szwarcman, Carlos Gomes, Gabby Nyirjesy, Blair Edwards, et al. Foundation models for generalist geospatial artificial intelligence. *arXiv preprint arXiv:2310.18660*, 2023.

Johannes Jakubik, Felix Yang, Benedikt Blumenstiel, Erik Scheurer, Rocco Sedona, Stefano Maurogiovanni, Jente Bosmans, Nikolaos Dionelis, Valerio Marsocci, Niklas Kopp, et al. TerraMind: Large-scale generative multimodality for earth observation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Konstantin Klemmer, Esther Rolf, Caleb Robinson, Lester Mackey, and Marc Rußwurm. SatCLIP: Global, general-purpose location embeddings with satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025a.

Konstantin Klemmer, Esther Rolf, Marc Russwurm, Gustau Camps-Valls, Mikolaj Czerkawski, Stefano Ermon, Alistair Francis, Nathan Jacobs, Hannah Rae Kerner, Lester Mackey, et al. Earth embeddings: Towards ai-centric representations of our planet. *EarthArXiv preprint*, 2025b. doi: 10.31223/X5HX9S.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Xuyang Li, Danfeng Hong, and Jocelyn Chanussot. S2MAE: A spatial-spectral pretraining foundation model for spectral remote sensing data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24088–24097, 2024.

Madeline C. Lisaius, Andrew Blake, Srinivasan Keshav, and Clement Atzberger. Using Barlow Twins to create representations from cloud-corrupted remote sensing time series. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:13162–13168, 2024.

Madeline C Lisaius, Srinivasan Keshav, Andrew Blake, and Clement Atzberger. Embedding-based crop type classification in the groundnut basin of senegal. *arXiv preprint arXiv:2601.16900*, 2026.

Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. RemoteCLIP: A vision language foundation model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 2024.

Junyuan Liu, Xinglei Wang, Zichao Zeng, Jiazhuang Feng, Quan Qin, Ilya Ilyankou, Guangsheng Dong, and Tao Cheng. CITYREP: A unified benchmark for urban representations across cities, tasks, and modalities. *arXiv preprint arXiv:2605.26036*, 2026.

Junwei Luo, Zhen Pang, Yongjun Zhang, Tingzhu Wang, Linlin Wang, Bo Dang, Jiangwei Lao, Jian Wang, Jingdong Chen, Yihua Tan, et al. SkySenseGPT: A fine-grained instruction tuning dataset and model for remote sensing vision-language understanding. *arXiv preprint arXiv:2406.10100*, 2024.

Jienan Lyu, Miao Yang, Jinchen Cai, Yiwen Hu, Guanyi Lu, Junhao Qiu, and Runmin Dong. Structure-semantic decoupled modulation of global geospatial embeddings for high-resolution remote sensing mapping. *arXiv preprint arXiv:2604.19591*, 2026.

Oscar Mañas, Alexandre Lacoste, Xavier Giró-i Nieto, David Vazquez, and Pau Rodríguez. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9394–9403, 2021. doi: 10.1109/ICCV48922.2021.00928.

Matías Mendieta, Boran Han, Xingjian Shi, Yi Zhu, and Chen Chen. Towards geospatial foundation models via continual pretraining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16760–16770, 2023. doi: 10.1109/ICCV51070.2023.01541.

Daniel B. Metcalfe, Emily Anders, Hanna Axén, et al. Gaps in tropical science from unrepresentative distribution of sampling and citation across natural terrestrial environments. *Nature Communications*, 2025. doi: 10.1038/s41467-025-67617-4.

Andrea Nascetti, Ritu Yadav, Kirill Brodt, Qixun Qu, Hongwei Fan, Yuri Shendryk, Isha Shah, and Christine Chung. Biomassters: A benchmark dataset for forest biomass estimation using multimodal satellite time-series. *Advances in Neural Information Processing Systems*, 36:20409–20420, 2023.

Mubashir Noman, Muzammal Naseer, Hisham Cholakkal, Rao Muhammad Anwar, Salman Khan, and Fahad Shahbaz Khan. Rethinking transformers pre-training for multi-spectral satellite imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27811–27819, 2024. doi: 10.1109/CVPR52733.2024.02627.

Yohann Perron, Guillaume Astruc, Nicolas Gonthier, Clement Mallet, and Loic Landrieu. Universat: Resolution- and modality-agnostic transformers for earth observation. *arXiv preprint arXiv:2606.23503*, 2026.

PyTorch Team. `torch.utils.flop_counter.FlopCounterMode`: per-operator flop accounting in pytorch. https://pytorch.org/docs/stable/torch.utils.flop_counter.html, 2024. Accessed 2026.

Xiaolei Qin, Di Wang, Jing Zhang, Fengxiang Wang, Xin Su, Bo Du, and Liangpei Zhang. TiMo: Spatiotemporal foundation model for satellite image time series, 2025. URL https://arxiv.org/abs/2505.08723.

Colorado J. Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-MAE: A scale-aware masked autoencoder for multiscale geospatial representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4065–4076, 2023. doi: 10.1109/ICCV51070.2023.00378.

Esther Rolf, Jonathan Proctor, Tamma Carleton, Ian Bolliger, Vaishaal Shankar, Miyabi Ishihara, Benjamin Recht, and Solomon Hsiang. A generalizable and accessible approach to machine learning with global satellite imagery. *Nature Communications*, 12(1):4392, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

Vivien Sainte Fare Garnot, Loic Landrieu, and Nesrine Chehata. Multi-modal temporal attention models for crop mapping from satellite time series. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2022. doi: https://doi.org/10.1016/j.isprsjprs.2022.03.012.

Johannes Schmude, Sujit Roy, Will Trojak, Johannes Jakubik, Daniel Salles Civitarese, Shraddha Singh, et al. Prithvi-WxC: Foundation model for weather and climate, 2024. URL https://arxiv.org/abs/2409.13598.

Michael J. Smith, Luke Fleming, and James E. Geach. EarthPT: a time series foundation model for Earth Observation, 2023. URL https://arxiv.org/abs/2309.07207.

Xian Sun, Peijin Wang, Wanxuan Lu, Zicong Zhu, Xiaonan Lu, Qibin He, Junxi Li, Xuee Rong, Zhujun Yang, Hao Chang, Qinglin He, Guang Yang, Ruiping Wang, Jiwen Lu, and Kun Fu. RingMo: A remote sensing foundation model with masked image modeling. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–22, 2023. doi: 10.1109/TGRS.2022.3194732.

Daniela Szwarcman, Sujit Roy, Paolo Fraccaro, Þorsteinn Eli Gislason, Benedikt Blumenstiel, Rinki Ghosal, et al. Prithvi-EO-2.0: A versatile multi-temporal foundation model for earth observation applications, 2025. URL https://arxiv.org/abs/2412.02732.

Maofeng Tang, Andrei Cozma, Konstantinos Georgiou, and Hairong Qi. Cross-scale MAE: A tale of multiscale exploitation in remote sensing. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:20054–20066, 2023.

Gabriel Tseng, Ivan Zvonkov, Catherine Lilian Nakalembe, and Hannah Kerner. Cropharvest: A global dataset for crop-type classification. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=JtjzUXPEaCu.

Gabriel Tseng, Ruben Cartuyvels, Ivan Zvonkov, Mirali Purohit, David Rolnick, and Hannah Kerner. Lightweight, pre-trained transformers for remote sensing timeseries, 2024. URL https://arxiv.org/abs/2304.14065.

Gabriel Tseng, Anthony Fuller, Marlena Reil, Henry Herzog, Patrick Beukema, Favyen Bastani, James R. Green, Evan Shelhamer, Hannah Kerner, and David Rolnick. Galileo: Learning global & local features of many remote sensing modalities, 2025. URL https://arxiv.org/abs/2502.09356.

Rikard Vinge, Isabelle Wittmann, Jannik Schneider, Michael Marszalek, Luis Gilch, Thomas Brunschwiler, and Conrad M. Albrecht. Neuco-bench: A novel benchmark framework for neural embeddings in earth observation, 2025. arXiv:2510.17914.

Yi Wang, Conrad M. Albrecht, and Xiao Xiang Zhu. Self-supervised vision transformers for joint SAR-optical representation learning, 2022. URL https://arxiv.org/abs/2204.05381.

Yuelei Wang, Ting Zhang, Liangjin Zhao, Lin Hu, Zhechao Wang, Ziqing Niu, Peirui Cheng, Kaiqiang Chen, Xuan Zeng, Zhirui Wang, Hongqi Wang, and Xian Sun. RingMo-lite: A remote sensing multi-task lightweight network with CNN-transformer hybrid framework, 2023. URL https://arxiv.org/abs/2309.09003.

Xinye Wanyan, Sachith Seneviratne, Shuchang Shen, and Michael Kirley. Extending global-local view alignment for self-supervised learning with remote sensing imagery, 2024. URL https://arxiv.org/abs/2303.06670.

Charith Wickrema, Eliza Mace, Hunter Brown, Heidys Cabrera, Nick Krall, Matthew O'Neill, Shivangi Sarkar, Lowell Weissman, Eric Hughes, and Guido Zarrella. Scaling remote sensing foundation models: Data domain tradeoffs at the peta-scale. *arXiv preprint arXiv:2512.23903*, 2025.

Nicholas Wright, John MA Duncan, J Nik Callow, Sally E Thompson, and Richard J George. Training sensor-agnostic deep learning models for remote sensing: Achieving state-of-the-art cloud and cloud shadow identification with OmniCloudMask. *Remote Sensing of Environment*, 322:114694, 2025.

Kang Wu, Yingying Zhang, Lixiang Ru, Bo Dang, Jiangwei Lao, Lei Yu, Junwei Luo, Zifan Zhu, Yue Sun, Jiahao Zhang, et al. A semantic-enhanced multi-modal remote sensing foundation model for earth observation. *Nature Machine Intelligence*, pp. 1–15, 2025a.

Kang Wu et al. A semantic-enhanced multi-modal remote sensing foundation model for Earth observation. *Nature Machine Intelligence*, 2025b.

Aoran Xiao, Weihao Xuan, Junjue Wang, Jiaxing Huang, Dacheng Tao, Shijian Lu, and Naoto Yokoya. Foundation models for remote sensing and earth observation: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 2025.

Zhitong Xiong, Yi Wang, Fahong Zhang, Adam J. Stewart, Joëlle Hanna, Damian Borth, Ioannis Papoutsis, Bertrand Le Saux, Gustau Camps-Valls, and Xiao Xiang Zhu. Neural plasticity-inspired multimodal foundation model for earth observation. *arXiv preprint arXiv:2403.15356*, 2024.

Fanglong Yao, Wanxuan Lu, Heming Yang, Liangyu Xu, Chenglong Liu, Leiyi Hu, Hongfeng Yu, Nayu Liu, Chubo Deng, Deke Tang, Changshuo Chen, Jiaqi Yu, Xian Sun, and Kun Fu. RingMo-Sense: Remote sensing foundation model for spatiotemporal prediction via spatiotemporal evolution disentangling. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–21, 2023. doi: 10.1109/TGRS.2023.3316166.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-supervised learning via redundancy reduction. *arXiv preprint arXiv:2103.03230*, 2021.

Linglin Zeng, Brian D. Wardlow, Daxiang Xiang, Shun Hu, and Deren Li. A review of vegetation phenological metrics extraction using time-series, multispectral satellite data. *Remote Sensing of Environment*, 237:111511, 2020.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Yingying Zhang, Lixiang Ru, Kang Wu, Lei Yu, Lei Liang, Yansheng Li, and Jingdong Chen. SkySense V2: A unified foundation model for multi-modal remote sensing. *arXiv preprint arXiv:2507.13812*, 2025.

Qi Zhu, Jiangwei Lao, Deyi Ji, Junwei Luo, Kang Wu, Yingying Zhang, Lixiang Ru, Jian Wang, Jingdong Chen, Ming Yang, et al. SkySense-O: Towards open-world remote sensing interpretation with vision-centric visual-language modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14733–14744, 2025.

A Design Desiderata and Extended Related Work

This appendix expands two parts of the main text that are stated compactly there: the five product criteria of Section 1 applied system by system, and the full catalogue of EO foundation models summarised in Section 2.

The five criteria. An embedding product is *analysis-ready* if a user can treat it as a geospatial data layer, with no raw-imagery processing and no GPU inference on their side; *transferrable* if one representation serves classification, regression, change detection, and high-resolution mapping across regions and workflows, rather than being tuned to a single benchmark (Klemmer et al., 2025b; Lyu et al., 2026); *reproducible* if the algorithms, training procedure, and weights can be inspected, compared, and extended rather than only their outputs downloaded; *economical* in the resources that actually limit EO adoption, namely labels, training and inference compute (and the associated carbon footprint), storage, I/O, and engineering time; and *adaptive* if it characterises the accuracy-versus-cost trade-off explicitly, so that users are not locked into one model size or one embedding dimensionality.

Table 1 scores the systems discussed in Sections 1 and 2 against these criteria. Remote-sensing foundation models are reproducible and transferrable but not analysis-ready, since the user must run the backbone. AlphaEarth (Brown et al., 2025) is analysis-ready at global scale but open-output only: the embeddings are public while the training procedure and weights are not (Hou et al., 2026). ESD (Chen et al., 2026) is analysis-ready and open, but at 30 m from Landsat/MODIS rather than 10 m from Sentinel-1/2. OlmoEarth (Herzog et al., 2026) is flexible in time window but computes embeddings per request, so it is not a precomputed archive. Tessera v1 (Feng et al., 2026) meets every criterion except adaptivity: one model, one dimension, one storage budget. Tessera v2 adds both missing degrees of freedom.

Table 1: Embedding products and foundation models against the five criteria of Section 1. ✓ = satisfied, ∼ = partially, × = not.

System	Analysis-ready	Transferrable	Reproducible	Economical	Adaptive
RSFMs (fine-tuned)	×	✓	✓	×	×
MOSAIKS	∼	∼	✓	✓	×
ESD	✓	∼	✓	✓	×
OlmoEarth	∼	✓	✓	∼	×
AlphaEarth	✓	✓	×	✓	×
Tessera v1	✓	✓	✓	✓	×
Tessera v2 (ours)	✓	✓	✓	✓	✓

Extended catalogue of EO foundation models. Section 2 cites a representative subset. The full set of spatial-backbone models pretrained on single-time, often cloud-filtered patches with contrastive or masked-image objectives includes (Mañas et al., 2021; Guo et al., 2024; Cong et al., 2022; Reed et al., 2023; Tang et al., 2023; Li et al., 2024; Noman et al., 2024; Wang et al., 2022; Wanyan et al., 2024), which have grown into broader families (Sun et al., 2023; Wang et al., 2023; Bastani et al., 2023; Schmude et al., 2024; Szwarcman et al., 2025; Mendieta et al., 2023; Han et al., 2024; Wu et al., 2025a; Luo et al., 2024; Zhu et al., 2025; Zhang et al., 2025). Multi-sensor and multi-resolution models fuse optical and SAR or ingest many sensors at once (Fuller et al., 2023; Yao et al., 2023; Xiong et al., 2024; Astruc et al., 2025; Perron et al., 2026; Houdré et al., 2026; Tseng et al., 2025).

Dataset frameworks and embedding archives. Beyond the models above, Major TOM (Francis & Czerkawski, 2024) standardises EO datasets through a global grid index and a shared metadata schema, releasing the near-global MajorTOM-Core Sentinel corpus; its embedding expansion (Czerkawski et al., 2024) adds dense patch-level embeddings extracted from pretrained backbones over that grid, across Sentinel-2 L1C/L2A and Sentinel-1 RTC. As with the embedding products in Section 2, each ships at a single fixed embedding specification per source.

B TRAINING-FLOPS DERIVATION

This appendix derives the training-FLOPs accounting that underlies the compute axis Equation (1), adopting the standard convention that one fused multiply–add corresponds to two FLOPs (Kaplan et al., 2020; Hoffmann et al., 2022). The scaling-study axis of Equation (1) is exactly this cost, evaluated at the mean per-view length $\bar{L} = 12$; the two coincide up to the $\mathcal{O}(DL^2)$ self-attention residual ($<1\%$; Appendix B.7), as collected in Appendix B.6.

B.1 NOTATION

- D: total number of d-pixels processed during training.
- L: per-modality temporal sequence length (number of tokens consumed by the temporal encoder per d-pixel).
- N_{enc}: total parameter count of all temporal encoder backbones, summed across modalities.
- N_{fus}: parameter count of the modality-fusion attention that combines the per-modality summary tokens into one joint representation.
- M: number of modality branches fused per d-pixel ($M \in \{2, 3\}$ across our configs—e.g. S2 + S1, or S2 + S1-asc + S1-desc).
- N_{red}: parameter count of the dimensionality reducer applied to the fused representation.
- N_{proj}: parameter count of the BARLOW TWINS projector alone ($= 5P^2$; the reducer N_{red} is counted separately).
- P: output dimension of the BARLOW TWINS projector.
- B: per-step batch size; B appears as an intermediate quantity and cancels in the final per-d-pixel expression.

B.2 PER-PARAMETER, PER-TOKEN FORWARD+BACKWARD COST

Consider a dense linear layer $y = xW^\top$ with $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and $n = d_{\text{in}}\, d_{\text{out}}$ parameters. For a single input token, the forward computation costs $2n$ FLOPs (one matrix–vector product); the backward pass with respect to the activation gradient, $\partial\mathcal{L}/\partial x = (\partial\mathcal{L}/\partial y)\, W$, costs $2n$ FLOPs; and the backward pass with respect to the weight gradient, $\partial\mathcal{L}/\partial W = (\partial\mathcal{L}/\partial y)^\top x$, costs another $2n$ FLOPs. The total cost of one forward+backward pass through this layer is

$$6n \quad \text{[FLOPs per parameter per token]}. \tag{3}$$

This accounting applies to the dense matrix multiplications that dominate transformer attention, feed-forward, and embedding layers; the $\mathcal{O}(d_{\text{model}})$ contributions of element-wise non-linearities and LayerNorm are subdominant and are absorbed in the residual budgeted in Appendix B.9. The $\mathcal{O}(L^2)$ self-attention quadratic correction is treated separately in Appendix B.7.

B.3 PER-d-PIXEL COST OF A SINGLE FORWARD+BACKWARD PASS

The pretraining model decomposes into two parts whose token-level activity differs.

Encoder. Each d-pixel contributes L tokens to the per-modality temporal encoder, and each token is routed through all N_{enc} encoder parameters. The encoder cost per d-pixel for one forward+backward pass is therefore

$$C^{(1)}_{\text{enc}} \;=\; 6\, N_{\text{enc}}\, L \quad \text{[FLOPs per } d\text{-pixel]}. \tag{4}$$

Modality fusion. Each modality is summarised into a single token, and a modality-fusion attention combines the M summary tokens into one joint representation. Because it operates on M tokens rather than the L temporal tokens, its cost per d-pixel is

$$C^{(1)}_{\text{fus}} \;=\; 6\, N_{\text{fus}}\, M \quad \text{[FLOPs per } d\text{-pixel]}. \tag{5}$$

Charging fusion at L tokens instead of M would over-count it by up to $\sim 20\%$ of C and is incorrect.

Post-fusion head. After per-modality attention pooling and the fusion Transformer, a single representation vector $h \in \mathbb{R}^{d_{\text{repr}}}$ per d-pixel remains; the dimensionality reducer and BARLOW TWINS projector then act on this single token, giving

$$C^{(1)}_{\text{head}} \;=\; 6\big(N_{\text{red}} + N_{\text{proj}}\big) \quad \text{[FLOPs per } d\text{-pixel]}. \tag{6}$$

Combined. Summing the three parts, the per-d-pixel cost of a single forward+backward pass is

$$C^{(1)} \;=\; 6\big(N_{\text{enc}}\, L + N_{\text{fus}}\, M + N_{\text{red}} + N_{\text{proj}}\big) \quad \text{[FLOPs per } d\text{-pixel]}. \tag{7}$$

B.4 THREE ENCODER PASSES PER STEP

Our BARLOW TWINS step uses mixup regularisation, which runs the encoder *three* times per d-pixel rather than twice: on the two augmented views $x^{(1)}, x^{(2)}$, and on a mixed view $x^{(m)} = \alpha\, x^{(1)} + (1-\alpha)\, \pi(x^{(2)})$, each a full forward+backward pass through backbone, fusion, reducer, and projector (Zbontar et al., 2021; Bandara et al., 2023). The fourth (shuffled) branch used by the mixup target is the permutation $\pi(z_2)$ of an already-computed projection and adds no pass. Tripling Equation (7) gives

$$C^{(\text{step})}_{\text{model}} \;=\; 18\big(N_{\text{enc}}\, L + N_{\text{fus}}\, M + N_{\text{red}} + N_{\text{proj}}\big) \quad \text{[FLOPs per } d\text{-pixel]}. \tag{8}$$

B.5 BARLOW TWINS LOSS: THE P^2 TERM

The BARLOW TWINS loss is computed from the cross-correlation matrix between the two batch-normalised projector outputs $\tilde{z}_1, \tilde{z}_2 \in \mathbb{R}^{B\times P}$:

$$\mathcal{L}_{\text{BT}} \;=\; \sum_i \big(c_{ii} - 1\big)^2 + \lambda \sum_{i\neq j} c_{ij}^2, \qquad c \;=\; \tfrac{1}{B}\, \tilde{z}_1^{\top}\, \tilde{z}_2 \in \mathbb{R}^{P\times P}. \tag{9}$$

Three matrix multiplications dominate the per-step cost:

- Forward: $c = \tilde{z}_1^{\top}\tilde{z}_2/B$, a $(P\times B)\cdot(B\times P)$ product, costs $2BP^2$ FLOPs.
- Backward in $\tilde{z}_1$: $\partial\mathcal{L}_{\text{BT}}/\partial\tilde{z}_1 = \tilde{z}_2\,(\partial\mathcal{L}_{\text{BT}}/\partial c)^{\top}/B$, a $(B\times P)\cdot(P\times P)$ product, costs $2BP^2$ FLOPs.
- Backward in $\tilde{z}_2$: the symmetric $(B\times P)\cdot(P\times P)$ product costs another $2BP^2$ FLOPs.

The $\mathcal{O}(BP)$ standardisation cost and the $\mathcal{O}(P^2)$ element-wise on/off-diagonal loss cost are subdominant. Summing and amortising over the B d-pixels in a step,

$$C^{(\text{step})}_{\text{BT}} \;=\; \frac{6\,B\,P^2}{B} \;=\; 6\,P^2 \quad \text{[FLOPs per } d\text{-pixel]}. \tag{10}$$

The cancellation of B confirms that batch size enters only as an amortisation factor in the per-d-pixel cost. The mixup regulariser adds a further loss on the mixed view, evaluated through a $B\times B$ Gram identity ($\|D^{\top}W\|_F^2 = \text{tr}(G_D G_W)$) that avoids materialising any $P\times P$ matrix; its cost is $\mathcal{O}(DBP)$ (linear, not quadratic, in P), a loss-side term separate from the dominant $18\times$ model term of Equation (8). We do not carry it onto the compute axis.

B.6 EXACT FORMULA AND RECONCILIATION WITH THE COMPUTE AXIS

Summing Equations (8) and (10) and multiplying by the total number of d-pixels processed gives the exact total training FLOPs,

$$\boxed{C_{\text{exact}} \;=\; 18\,D\big(N_{\text{enc}}\, L + N_{\text{fus}}\, M + N_{\text{red}} + N_{\text{proj}}\big) \;+\; 6\,D\,P^2.} \tag{11}$$

The compute axis. The scaling-study compute axis Equation (1), which sets the x-axis of Figure 2 and the iso-FLOP buckets, is exactly Equation (11) evaluated at the mean per-view token length $\bar{L} = 12$: the augmentation feeds $L \in \{8, 16\}$ each step, so $\mathbb{E}[L] = 12$ for every run. All four model terms—the $18\times$ encoder, modality-fusion, reducer, and projector contributions—and the $6DP^2$ BARLOW TWINS term are carried onto the axis; the $6DP^2$ term is retained because it reaches up

to $\sim$6% of C (widest projector, smallest encoder) and $\sim$2% at the largest encoders. The terms *not* carried onto the axis are the $\mathcal{O}(DL^2)$ self-attention cost of Appendix B.7, below 1% of C for every swept configuration ($L \leq 16$), and the mixup Gram regulariser, a loss-side term (Appendix B.5). Up to that residual the axis is training FLOPs, and all reported exponents are exponents in training compute.

B.7 Self-attention Quadratic Correction

Self-attention introduces an $\mathcal{O}(L^2)$ contribution that is not captured by the per-parameter, per-token accounting of Appendix B.2, because the $QK^\top$ and $\mathrm{softmax}(QK^\top)V$ operations involve token–token interactions whose cost scales with L^2 rather than with $L \cdot \mathrm{params}$. For a single transformer layer with model dimension d_{model}, the forward+backward attention quadratic cost per d-pixel is $12\, L^2\, d_{\mathrm{model}}$ FLOPs. With n_{layers} layers, n_{mod} modalities, and the three encoder passes per step (Appendix B.4), the total contribution is

$$C_{\mathrm{attn}} \;=\; 36\, D\, L^2\, d_{\mathrm{model}}\, n_{\mathrm{layers}}\, n_{\mathrm{mod}}. \tag{12}$$

For $L \leq 16$ across all swept configurations, this term contributes less than 1% of C_{train} (Appendix B.9), and we omit it from Equations (1) and (11).

B.8 Inference-FLOPs Accounting and the GPU-Year Fit

The pretraining-FLOPs formula derived above is specific to a mixup Barlow Twins training step (three encoder passes, forward and backward, plus the P^2 cross-correlation contribution). Deployment inference is simpler: the model is evaluated once per d-pixel, in forward mode only, and the projector is discarded. The per-d-pixel inference FLOPs accordingly reduce to

$$C_{\mathrm{inf}}^{(\text{1-pixel})} \;=\; 2\big(N_{\mathrm{enc}}\, L_{\mathrm{inf}} + N_{\mathrm{fus}}\, M + N_{\mathrm{red}}\big) \quad \text{[FLOPs per } d\text{-pixel]}, \tag{13}$$

where L_{inf} is the annual inference sequence length (the full observation stack processed at deployment, distinct from the subsampled training $\bar{L}$), N_{red} is the dimensionality reducer, the projector is not used, and the factor of 2 replaces the 6 of training since only the forward pass is evaluated. Multiplying by the number of $10\,\mathrm{m}$ d-pixels on land, $N_{d-pixel} \approx 1.5\times10^{12}$ (Earth's $\approx 1.49\times10^{8}\,\mathrm{km}^2$ of land surface at $10\,\mathrm{m}$ resolution, i.e. $100\,\mathrm{m}^2$ per pixel), gives the total inference FLOPs per global annual pass plotted on the colour scale of Figure 7.

From total inference FLOPs to GPU-years. Converting total FLOPs into single-H100-year-equivalents uses the H100 SXM dense `bf16` tensor-core peak throughput, $T_{H100} \approx 9.9\times10^{14}$ FLOPs/s ($\approx$990 TFLOP/s). One H100-year of compute is then $T_{H100} \times (3.156\times10^{7}\,\mathrm{s}) \approx 3.1\times10^{22}$ FLOPs, the unit of the y-axis in Figure 7(a).

Empirical fit. Across the encoder-size range spanned by our students and the teacher, the resulting cost is well-approximated by a power law in the encoder parameter count N (in millions), $\mathrm{GPUy}(N) \approx 0.041 \cdot N^{1.03}$ (Equation (32)). The exponent of approximately unity reflects the fact that, in this family, inference cost is dominated by the encoder term $N_{\mathrm{enc}}\, L_{\mathrm{inf}}$ rather than by $N_{\mathrm{fus}}\, M$ or N_{red}. The coefficient follows from Equation (13) with $N_{d-pixel} = 1.5\times10^{12}$ land d-pixels and the H100-year unit above; substituting these, 0.041 corresponds to an annual inference length of $L_{\mathrm{inf}} \approx 424$ tokens summed over the M modalities. The fit accounts for encoder inference, but this is the dominant term: in the full embedding-as-data pipeline, data preparation (temporal sampling and stacking, disk I/O, and normalisation/fusion/quantisation) adds only $\sim$6% on top of the encoder forward pass (Figure 7b), so the GPUy fit is a tight proxy for end-to-end cost.

B.9 Empirical Validation

We validate Equation (11) numerically against `torch.utils.flop_counter.FlopCounterMode` (PyTorch Team, 2024), which adopts the same one-multiply-add-equals-two-FLOPs convention, across a range of encoder configurations, sequence lengths, and batch sizes. Measured on the full mixup training step (all three encoder passes and the modality-fusion attention), the formula matches the counted FLOPs to within 2.2%; dropping the fusion term $N_{\mathrm{fus}}\, M$ leaves a shortfall that *shrinks* as L grows (2.6% at the shortest sequences, 1.3% at the longest).

Because the fusion cost $N_{\text{fus}}\,M$ is itself independent of L while the encoder term grows with L, this decreasing fraction is the signature of fusion—not of the $\mathcal{O}(DL^2)$ attention term—as the leading correction at small L; the attention term, which *grows* with L, overtakes it only at larger L than we sweep. The per-d-pixel cost is invariant to batch size, confirming that B enters only as an amortisation factor.

C Mix-up Regularisation Derivation

This appendix derives the mix-up loss of Section 4 and Equation (30) following Bandara et al. (2023) and Feng et al. (2026).

C.1 Motivation

Barlow Twins enforces invariance under view augmentation: two augmented views Y_A, Y_B of the same d-pixel should produce near-identical and dimension-decorrelated embeddings. Under sparse temporal sampling, this invariance is sometimes too easy to satisfy: with only L valid timesteps drawn per view, the network can find shortcut representations that ignore physical phenology and rely on trivial temporal-summary statistics. The mix-up regulariser prevents this collapse by imposing a stronger consistency property: *linear interpolation in the input space should correspond to linear interpolation in the embedding's cross-correlation space.* If $z = f(x)$ preserves physical structure, then the cross-correlations of $f(\alpha x_1 + (1-\alpha)x_2)$ with $f(x_1)$ and $f(x_2)$ should obey a convex-combination identity.

C.2 Construction

For each training step we have:

- two augmented views Y_A, Y_B of size (B, L, C) (batch $\times$ time $\times$ channels);
- a shuffled view $Y_S = \text{Shuffle}(Y_B)$, where Shuffle permutes the batch dimension and breaks the correspondence between (Y_A, Y_S) pairs;
- a mix coefficient $\alpha_{\text{mix}} \sim \mathcal{U}(0,1)$ sampled per training step;
- a mixed view $Y_M = \alpha_{\text{mix}}\, Y_A + (1-\alpha_{\text{mix}})\, Y_S$.

Passing each view through the dual-branch encoder and projector gives batch-normalised embeddings $Z_A, Z_S, Z_M \in \mathbb{R}^{B\times P}$.

C.3 Cross-correlation Identities Under Linear Consistency

The mix-up regulariser is built around two cross-correlations of the mixed embedding with the endpoint embeddings:

$$C^{MA} \;=\; \tfrac{1}{B}\, Z_M^\top Z_A \;\in\; \mathbb{R}^{P\times P}, \qquad C^{MS} \;=\; \tfrac{1}{B}\, Z_M^\top Z_S \;\in\; \mathbb{R}^{P\times P}, \tag{14}$$

normalised by B exactly as the Barlow Twins cross-correlation $c = \frac{1}{B}\tilde{z}_1^\top \tilde{z}_2$ of Equation (29) and Appendix B.5, so that $\mathcal{L}_{\text{MIX}}$, and hence the weight λ_{mix}, are independent of batch size. If the encoder f respects linear interpolation exactly, then

$$Z_M \;\approx\; \alpha_{\text{mix}}\, Z_A + (1-\alpha_{\text{mix}})\, Z_S, \tag{15}$$

in which case substitution into C^{MA} and C^{MS} yields

$$C^{MA}_{\text{target}} \;=\; \tfrac{1}{B}\big(\alpha_{\text{mix}}\, Z_A^\top Z_A + (1-\alpha_{\text{mix}})\, Z_S^\top Z_A\big), \tag{16}$$

$$C^{MS}_{\text{target}} \;=\; \tfrac{1}{B}\big(\alpha_{\text{mix}}\, Z_A^\top Z_S + (1-\alpha_{\text{mix}})\, Z_S^\top Z_S\big). \tag{17}$$

These are the convex-combination targets that the mixed embedding should match if the linear-consistency property holds.

C.4 Loss

The mix-up loss penalises Frobenius-norm deviations from the interpolation-predicted targets,

$$\mathcal{L}_{\text{MIX}} \;=\; \tfrac{1}{2}\big\|C^{MA}-C^{MA}_{\text{target}}\big\|_F^2 + \tfrac{1}{2}\big\|C^{MS}-C^{MS}_{\text{target}}\big\|_F^2, \tag{18}$$

which is Equation (30). The total pretraining objective combines Equations (18) and (29),

$$\mathcal{L}_{\text{total}} \;=\; \mathcal{L}_{\text{BT}} + \lambda_{\text{mix}}\,\mathcal{L}_{\text{MIX}}, \qquad \lambda_{\text{mix}}=1.0, \tag{19}$$

with λ_{mix} following Bandara et al. (2023).

C.5 Why Two Pairs Rather Than One

A simpler variant would penalise $Z_M \approx \alpha_{\text{mix}} Z_A + (1-\alpha_{\text{mix}}) Z_S$ directly in the embedding space. The cross-correlation form of Equation (18) has two advantages over this primal formulation: (i) it is invariant to global orthogonal transformations of the embedding, consistent with the rotation symmetry of Barlow Twins analysed in Appendix V; and (ii) by constraining cross-correlations of the mixed view against *both* endpoint embeddings, it provides two independent supervisory signals per training step. The regulariser vanishes at the endpoints $\alpha_{\text{mix}} \in \{0, 1\}$, where the mixed view coincides with Y_S or Y_A and both cross-correlations equal their targets identically; it is informative only for intermediate α_{mix}, which is why the coefficient is resampled every step.

C.6 Empirical Importance

Tessera v1's ablations on Austrian Crop classification report that removing mix-up alone drops validation F1 by -11.1 points (from 77.3 to 66.2) and RankMe by -0.106; removing both global shuffling and mix-up drops F1 by -14.7 points (Feng et al., 2026). We retain both regularisers unchanged in Tessera v2.

D Adaptive Bucket Sampling

This section gives the construction summarised in Section 4 together with a self-contained reference implementation.

Bucket checkpoints. We define an arithmetic ladder of allowed sequence lengths,

$$\mathcal{B} \;=\; \big\{\, B_n = a_0 + n\,\Delta \;:\; n = 0, 1, 2, \dots \big\}, \qquad a_0 = \Delta = 16, \tag{20}$$

i.e., $\mathcal{B} = \{16, 32, 48, 64, 80, 96, \dots\}$. These ladder values are the only sequence lengths the encoder is asked to consume at inference; the shortest bucket coincides with the longest pretraining length ($L=16$).

Selecting a bucket. For a pixel with k valid observations of a given modality, we pick the smallest bucket that fits:

$$B^\star(k) \;=\; \min\big\{\, B \in \mathcal{B} \;:\; B \geq k \,\big\}. \tag{21}$$

Letting $\boldsymbol{o} \;=\; (o_0, o_1, \dots, o_{k-1})$ denote the k valid observations in temporal order, the encoder consumes a sequence $\tilde{\boldsymbol{o}} \in \mathbb{R}^{B^\star}$ constructed as follows.

Sequence construction. The first k positions of $\tilde{\boldsymbol{o}}$ hold the unique real observations in time order; the remaining $M = B^\star - k$ positions are filled by *midpoint resampling* from $\boldsymbol{o}$. We partition the index set $\{0, \dots, k-1\}$ into M equally-spaced groups of mean size k/M and copy the midpoint of each group into one of the residual slots. The source index for the j-th residual slot ($j = 0, 1, \dots, M-1$) is

$$s_j \;=\; \left\lfloor \tfrac{(2j+1)\,k}{2\,M} \right\rfloor, \tag{22}$$

and $\tilde{\boldsymbol{o}}$ is assembled by

$$\tilde{\boldsymbol{o}}_i \;=\; \begin{cases} o_i & 0 \leq i \leq k-1, \\ o_{s_{i-k}} & k \leq i \leq B^\star - 1. \end{cases} \tag{23}$$

The choice of s_j in Equation (22) places duplicates at uniformly spaced anchor points along the year, so that resampled slots are statistically representative of the full temporal range rather than concentrated near its endpoints. When $k = B^\star$ no residual slots exist and $\tilde{\boldsymbol{o}} = \boldsymbol{o}$ exactly.

Why it helps. Every valid observation enters the sequence at its true temporal index, so the encoder sees the full phenology at every pixel; pixels with more observations get longer buckets, so high-data regions are not throttled to a fixed ceiling. Replication is confined to the residual tail and is uniformly distributed in time, which keeps the empirical sample distribution close to the training distribution. Across pixels, the bucket assignment partitions a batch into a small number of fixed-length groups, on which the encoder runs efficiently in parallel.

Reference implementation. Algorithm 1 implements the sampler; it runs once per pixel per modality at inference, and results are concatenated across pixels with matching bucket size to form fixed-length encoder batches.

Algorithm 1 ADAPTIVEBUCKETSAMPLE — inference-time fixed-length packing of valid observations.

Require: Valid observations $\boldsymbol{o} = (o_0, \ldots, o_{k-1})$ in temporal order; bucket ladder $\mathcal{B} = \{a_0 + n\Delta : n \geq 0\}$.
Ensure: Padded sequence $\tilde{\boldsymbol{o}}$ of length $B^\star$.

$B^\star \leftarrow \min\{B \in \mathcal{B} : B \geq k\}$ {smallest bucket fitting k}
Allocate $\tilde{\boldsymbol{o}}$ of length $B^\star$
for $i = 0$ **to** $k - 1$ **do**
  $\tilde{\boldsymbol{o}}_i \leftarrow o_i$ {unique real observations}
end for
$M \leftarrow B^\star - k$ {number of duplicates}
for $j = 0$ **to** $M - 1$ **do**
  $s_j \leftarrow \lfloor (2j + 1)\, k \,/\, (2M) \rfloor$ {midpoint of j-th group}
  $\tilde{\boldsymbol{o}}_{k+j} \leftarrow o_{s_j}$
end for
return $\tilde{\boldsymbol{o}}$

Worked examples. *Example 1 (Sentinel-2, $k = 40$).* The smallest bucket ≥ 40 is $B^\star = 48$, so $M = 8$ duplicates are needed. The source indices from Equation (22) are $\{2, 7, 12, 17, 22, 27, 32, 37\}$, evenly spaced by $\Delta s = 5$, i.e., every fifth real observation starting from index 2. *Example 2 (Sentinel-1, $k = 30$).* $B^\star = 32$, $M = 2$; the source indices are $\{7, 22\}$—the midpoints of the two halves $\{0, \ldots, 14\}$ and $\{15, \ldots, 29\}$. Both examples instantiate the same general rule.

E INFERENCE-TIME TEMPORAL WINDOWS

The embedding products released under the embedding-as-data paradigm summarise a full calendar year, yet the encoder itself imposes no fixed horizon. Each modality branch aggregates its observations by length-agnostic attention pooling (Section 4), so the same weights map any temporal window, from a few dates to a full year, onto a single embedding. Pretraining, however, exposes the model only to annual inputs. Whether an embedding computed over a shorter window is still useful downstream is therefore an empirical question rather than an architectural one.

The OpenET evapotranspiration ensemble, one of the 15 ALPHAEARTH suite tasks (Appendix L), shows that the answer is task-dependent, and that a sub-annual window can be preferable. Evaluated with an annual window, the Tessera v2-2B-L model scores 0.51 on this task, behind ALPHAEARTH at 0.58. Recomputing the embedding over a monthly window, with no change to the model or its weights, raises the score to 0.69 (Table 2). We attribute the difference to the strong intra-annual dynamics of evapotranspiration, which a single yearly summary averages out but a monthly window retains. This does not imply that shorter windows are uniformly better. It shows that the temporal support best matched to a target variable need not coincide with the annual product, even for a model trained only on annual inputs.

The distinction affects how the embeddings are served. The precomputed products fix the window to the calendar year (1 January to 31 December), which suits variables with slow annual dynamics and keeps the released fields comparable across years. Applications whose signal concentrates at sub-annual scales, or that require a window with an arbitrary start date or a non-annual length, are

Table 2: **Effect of the inference window on the OpenET ensemble.** The Tessera v2-2B-L model evaluated at an annual and a monthly window, with ALPHAEARTH for reference. Scores use the same chance-adjusted convention as the rest of our benchmarks; higher is better.

Model (inference window)	OpenET ensemble
ALPHAEARTH (annual)	0.58
Tessera v2-2B-L (annual window)	0.51
Tessera v2-2B-L (monthly window)	0.69

better served by running the released encoder over the relevant date range, at the additional inference cost quantified in Appendix U.

F ARCHITECTURE SWEEP DETAILS

Prior to the scaling sweep we fix the structural choices of the model family with a one-axis-at-a-time architecture sweep: each axis is varied while the other axes are held at the finally selected configuration (scheme A), and every candidate is evaluated on three downstream tasks, Canada coarse crop-type classification (balanced accuracy, ↑), ASTER GED regression (R^2, ↑), and LCMAP land-use-change classification (balanced accuracy, ↑; a downsampled version of the full dataset). All three metrics lie on a $[0, 1]$ higher-is-better scale, and the composite is their arithmetic mean. Each axis was swept as its own round with its own baseline run (its own data shard, seed, and short schedule), so the selected configuration's composite is not a single shared number but varies by about ± 0.01 between rounds; comparisons are therefore meaningful within an axis, not across axes, and within-axis gaps of $\lesssim 0.01$ are treated as ties. Table 3 reports the full sweep. On seven of the nine model axes (the MATRYOSHKA-layout block is discussed separately below) the selected scheme-A configuration is also the highest-composite candidate, and two axes are exceptions. On temporal aggregation, GRU pooling edges out attention pooling by a hair (0.725 vs. 0.718), yet we still select attention pooling: it keeps the encoder a single all-transformer stack that is simpler to scale, and its pooling is faster at inference than a sequential recurrent pass over the time series. On cloud handling, the default SCL mask and OmniCloudMask are within noise of each other on this three-task probe, and OmniCloudMask is marginally ahead, but we keep the SCL mask because it ships with the Sentinel-2 L2A product and needs no extra masking model at inference. The time-series-length axis is selected as a mixture rather than a single value: the deployed model samples $L \in \{8, 16\}$ per view during pretraining (mean length $\bar{L} = 12$; Appendix B), so both $L = 8$ and $L = 16$ are marked in bold, and they are also the two strongest single-length probes (0.711 and 0.708), so this axis retains the top candidate rather than trading it away. The final MATRYOSHKA-layout axis is different: it asks how to introduce nested prefixes *during pretraining*, and every such variant underperforms—the best of them (four prefix projectors with averaged BARLOW TWINS, composite 0.651) still falls well short of the other axes' selected scores. This is why scheme A introduces *no* prefix nesting during pretraining and defers MATRYOSHKA learning to the distillation stage, consistent with the rotation-symmetry analysis of Appendix V; the distillation-stage alternative is evaluated in Appendix W.2, not here.

Table 3: **Architecture sweep.** Each axis is varied with all other axes held at the selected scheme-A configuration; candidates are scored on three downstream tasks and their averaged composite. Bold marks the selected (scheme-A) candidate per axis. The MATRYOSHKA-layout axis lists pretraining-time nesting variants only, none of which is adopted (Appendix V).

Candidate	Canada BA ↑	ASTER GED R^2 ↑	LCMAP-LUC BA ↑	Composite ↑
Encoder architecture				
CNN–GRU	0.557	0.781	0.747	0.695
CNN–Transformer hybrid	0.584	0.781	0.744	0.703
Transformer (ReLU FFN)	0.585	0.805	0.767	**0.719**
Transformer + QK-Norm	0.586	0.802	0.748	0.712
Transformer + SwiGLU FFN	0.574	0.798	0.746	0.706
Input modalities				
Sentinel-2 + Sentinel-1	0.585	0.784	0.754	**0.708**
Sentinel-2 only	0.550	0.752	0.705	0.669
Sentinel-2 + Sentinel-1 + Landsat	0.481	0.693	0.650	0.608
Temporal position encoding				
None (index order)	0.422	0.561	0.538	0.507
Learned positional embedding	0.547	0.737	0.690	0.658
Sinusoidal DoY encoding	0.561	0.786	0.749	**0.699**
Temporal aggregation				
Mean pooling	0.549	0.762	0.732	0.681
GRU pooling	0.592	0.810	0.772	0.725
GRU + attention (custom)	0.568	0.785	0.765	0.706
Attention pooling	0.586	0.807	0.761	**0.718**
LTAE-style pooling	0.581	0.796	0.765	0.714
SAR (Sentinel-1) modality handling				

Continued on next page

Table 3, continued from previous page

Candidate	Canada BA ↑	ASTER GED R^2 ↑	LCMAP-LUC BA ↑	Composite ↑
Separate ascending/descending encoders	0.567	0.783	0.753	0.701
Unified Sentinel-1 encoder (asc+desc)	0.569	0.786	0.757	**0.704**
Fusion layer				
Summation	0.563	0.767	0.737	0.689
Concatenation + MLP	0.565	0.775	0.748	0.696
Cross-attention fusion	0.574	0.788	0.750	0.704
Fusion Transformer	0.577	0.793	0.766	**0.712**
Projector form				
Linear projector	0.474	0.626	0.613	0.571
2-layer BN-MLP	0.544	0.743	0.722	0.670
Deep BN-MLP (4 hidden layers, 4096-d)	0.566	0.800	0.744	**0.703**
Wider deep MLP (8192-d)	0.559	0.790	0.739	0.696
Cloud handling				
Default SCL cloud mask	0.559	0.772	0.751	**0.694**
OmniCloudMask (Wright et al., 2025)	0.558	0.775	0.758	0.697
Time-series length				
$L = 8$	0.582	0.790	0.761	**0.711**
$L = 16$	0.570	0.790	0.764	**0.708**
$L = 32$	0.579	0.789	0.729	0.699
$L = 48$	0.553	0.778	0.739	0.690
$L = 64$	0.551	0.773	0.725	0.683
MATRYOSHKA *layout (pretraining-time nesting variants; none adopted)*				
Naïve M-BT: 4 prefix projectors, averaged	0.537	0.722	0.693	0.651
Naïve M-BT: shared-trunk, summed	0.505	0.701	0.653	0.620
Naïve M-BT: multi-path summed	0.476	0.665	0.620	0.587
Naïve M-BT: segmented projectors	0.504	0.662	0.618	0.595
Repr-level decorrelation (prefix)	0.459	0.587	0.572	0.539
Repr-level decorrelation (segmented)	0.458	0.608	0.604	0.557

Three of the axes are worth a word beyond their composite scores: in each, a qualitative effect rather than the number drove the final choice.

Adding Landsat as a third optical source lowered downstream accuracy in every configuration we tried, despite the extra spectral coverage it nominally brings. We resampled Landsat onto the 10 m Sentinel-2 grid before fusion and tested both nearest-neighbour and cubic interpolation; the two gave the same result, so the drop is not an artefact of the resampling kernel. Dropping Landsat's coarse thermal-infrared bands recovered part of the loss, but no variant that kept Landsat reached the score of Sentinel-1/2 alone. The most likely cause is resolution mismatch: once upsampled to 10 m, the Landsat pixels add little surface information that Sentinel-2 does not already carry, and they contribute noise the encoder has to fit around.

The default SCL cloud mask is unreliable in practice. It lets thin cirrus through, and it occasionally marks bright but valid surfaces as cloud or haze. We therefore built a second copy of the training data masked with OmniCloudMask (Wright et al., 2025), a learned cloud-detection model, expecting the cleaner supervision to help. Downstream the two versions were almost indistinguishable. The likely reason is that the encoder already discounts cloud-contaminated observations without being told to: cloudy pixels sit far out in reflectance space, off the distribution of clear ones, and the temporal model can down-weight them on its own. Since the better mask bought no measurable accuracy and OmniCloudMask adds an inference pass per tile, we keep SCL for the production v2 embeddings.

A single unified Sentinel-1 encoder and separate ascending/descending encoders scored within noise of each other. On physical grounds the separate treatment is easier to defend, since the two orbit geometries view the surface from different angles and carry different information. We still adopt the unified encoder, for a reason the composite does not reflect: in local embedding-generation runs it visibly reduced the tiling seams that appear where adjacent ascending and descending passes are encoded apart.

G FULL SCALING-LAW SETUP

This appendix specifies the scaling study of Section 3 in full: the compute proxy, the sweep grid, the evaluation suite, and the fitting procedure. Every number is reproducible from the released 395-run table.

Compute estimation. Each run's training compute is the exact total training FLOPs of Equation (11), evaluated from its configuration as

$$C \;=\; \kappa\, D\,\big(N_{\text{enc}}\,\bar{L} + N_{\text{fus}}\,M + N_{\text{red}} + N_{\text{proj}}\big) \;+\; 6\,D\,P^2, \qquad \kappa = 18, \quad \bar{L} = 12, \tag{24}$$

with encoder parameters N_{enc}, fusion parameters N_{fus} over M modality tokens, dimensionality reducer N_{red}, projector parameters $N_{\text{proj}} = (d_{\text{proj}}{+}1)\,P^2 = 5P^2$ (projector depth $d_{\text{proj}}{=}4$, width P; all five linear layers are charged at P^2, so the narrower input layer is slightly over-counted, a few percent of C at the smallest encoders and negligible at the largest), and D the number of d-pixels processed. The constant $\kappa = 2 \times 3 \times 3 = 18$ collects the two FLOPs per multiply–add and the threefold forward-plus-backward cost (together the textbook 6, Appendix B.2) times the *three* encoder passes per step—the two BARLOW TWINS augmentation views and the mixup view (Appendices B.4 and B.6); it matches the prefactor of Equation (1). The token length $\bar{L} = 12$ is the mean per-view length actually fed to the encoder each step (the augmentation samples $L \in \{8, 16\}$, so $\mathbb{E}[L] = 12$; Section 4). The $6DP^2$ BARLOW TWINS cross-correlation term is retained (κ and $\bar{L}$ enter the encoder term only); the sole omission is the $\mathcal{O}(DL^2)$ attention term ($< 1\%$ at $L \leq 16$), leaving C equal to training FLOPs to within that residual, which matches a `FlopCounterMode` reference to within 2.2%. Across the sweep the nine isoFLOP bucket centres span $1.3{\times}10^{15}$ to $7.3{\times}10^{18}$ FLOPs (the legend values of Figure 2b–d).

Sweep grid. We pretrain 395 models on a four-axis isoFLOP grid: encoder size (16 widths, 7–278 M), projector width ($P \in \{2048, 4096, 8192, 16384\}$, depth 4), training data (0.03–3,202 M d-pixels), and compute (nine log-spaced isoFLOP slices, centres 1.3×10^{15}–7.3×10^{18} FLOPs). Table 4 lists the slices and the configurations placed on each. Within a slice, the *encoder* sub-sweep fixes $P{=}4096$ and varies width, the *projector* sub-sweep fixes the encoder and varies P, and the *data* sub-sweep walks D off the isoFLOP curve at fixed N. The encoder and projector compute-optimal fits of Section 3.3 use the within-slice vertices; the data exponent is read from the same isoFLOP vertices, where $D^\star$ follows from $N^\star$ and the compute identity. The off-curve data sub-sweep is shown only to illustrate the score's sensitivity to D around the optimum and does not enter the $D^\star(C)$ fit.

Table 4: **The nine isoFLOP slices of the scaling sweep** (the viridis colour scale of Figure 2b–d). Encoder widths are drawn from $\{7, 11, 17, 21, 27, 35, 48, 60, 76, 95, 120, 151, 175, 194, 230, 278\}$ M.

Slice	C centre (FLOPs)	#runs	enc. sizes (M)	data range (M)
1	1.3×10^{15}	17	7–278 (9)	0.03–0.9
2	3.7×10^{15}	39	7–278 (11)	0.05–2.5
3	1.1×10^{16}	58	7–278 (14)	0.11–11.5
4	3.2×10^{16}	48	7–278 (15)	0.32–33.4
5	9.6×10^{16}	52	7–278 (16)	0.95–76.9
6	2.8×10^{17}	57	7–278 (16)	2.9–177
7	8.3×10^{17}	44	7–278 (10)	8.1–941
8	2.5×10^{18}	47	7–278 (16)	32–2,167
9	7.3×10^{18}	33	7–278 (14)	98–3,202

Task suite. Every run is evaluated on the same 15-task ALPHAEARTH suite used for the headline benchmark (Appendix L), spanning classification (land cover, land use, crop type, tree genera, oil palm), change detection, and regression (evapotranspiration, surface emissivity) drawn from 10 source datasets. For comparability across heterogeneous tasks we use chance-adjusted metrics: classification reports $(\text{balanced accuracy} - 1/K)/(1 - 1/K)$ for K classes and regression reports $\max(0, R^2)$. The 15 per-task scores are averaged with *uniform weight* into the single composite downstream score that is the y-axis of Figure 2a–c and the target of the isoFLOP fits (range -0.20 to 0.54 across the sweep). The per-run normalised BARLOW TWINS validation loss is logged but *not* optimised against (range 0.004 to 0.046).

Fitting procedure. Within each isoFLOP slice we fit a quadratic to the downstream score as a function of $\log N$ (encoder or projector); the vertex is the compute-optimal size at that compute level. A log–log least-squares fit through the per-slice vertices gives the power laws of Section 3.3. The encoder and data laws are fit on all nine slices (compute-optimal = the vertex of the within-slice quadratic). The projector optimum is only well-resolved once the downstream score clears the noise floor, so the projector law is fit on the mid-to-high compute slices rather than all nine. Uncertainty is a 2000-resample bootstrap over the per-slice vertices; robustness is assessed by leave-one-slice-out (LOBO) refits. The resulting exponents, confidence intervals, and LOBO spreads are reported in Table 20: the encoder exponent is 0.36 and the projector exponent is statistically indistinguishable from zero. The data exponent is not fit independently: on the isoFLOP vertices, fixing $N^\star$ pins $D^\star$ through $C \propto N \cdot D$, so 0.36 and 0.63 sum to ≈ 1 (here 0.99; the N-independent projector term makes this approximate rather than exact). The substantive quantity is therefore the encoder exponent itself; the sum matches the Hoffmann et al. (2022) balance by construction. Pretraining loss and downstream score are weakly anti-correlated over all 395 runs (Pearson $r = -0.18$, Spearman $\rho = -0.16$).

A data-heavy allocation. While the exponent *sum* is fixed by the isoFLOP constraint, the *split* is not, and it is informative. The compute-optimal allocation is markedly data-heavy: at the fitted exponents, doubling the budget calls for about $2^{0.36} \approx 1.28\times$ the encoder but $2^{0.63} \approx 1.55\times$ the data, so the compute-optimal frontier grows data faster than parameters ($D^\star \propto N^{\star\, 0.63/0.36}$, exponent ≈ 1.75). This departs from the roughly balanced parameter/data split Hoffmann et al. (2022) report for language. We have no first-principles account, but the likely cause is informational: a single d-pixel carries far less independent signal than a text token, since cloud cover and repeated orbital passes make consecutive observations highly redundant, so useful supervision accrues mainly through data volume rather than encoder capacity. It is also the tilt behind the enlarged teacher's correspondingly large training corpus (Section 4).

Large-model validation. Guided by the fitted law we train three teachers at 0.5B/1B/2B encoders on the compute-optimal data prescribed by $D^\star(C)$ (the 2B teacher uses $\sim$14B d-pixels). Evaluated on the same suite they score $0.572/0.596/0.608$, landing on the extrapolated small-model curve (Figure 2f) and surpassing Tessera v1 and AlphaEarth; distillation to compact students (Figure 2f, inset) costs ≤ 0.015 downstream score.

Within-bucket loss–score correlation. The pooled Pearson $r = -0.18$ of Section 3.2 mixes runs across four orders of magnitude of compute and four projector widths, which dilutes any within-budget relationship. The quantity that actually governs model selection is the loss–score correlation *inside* a fixed compute budget. Table 5 reports the Spearman ρ within each of the nine isoFLOP buckets: the values scatter around zero (mean -0.08, median -0.13, range $[-0.25, +0.07]$), and no bucket reaches a correlation strong enough to rank runs reliably by loss. Controlling for compute therefore does not uncover a hidden predictive signal; the loss is a weak proxy for downstream utility at every budget, not only in aggregate. **F1** rests on this within-budget statement, so removing the pooling confound strengthens it.

Table 5: **Loss–score Spearman ρ within each isoFLOP bucket**, over the runs in each bucket of Table 4. A value near zero means the Barlow Twins loss does not rank runs by downstream score at that compute budget; every bucket is close to zero.

Bucket	C centre (FLOPs)	n	Spearman ρ
1	1.3×10^{15}	17	-0.245
2	3.7×10^{15}	39	-0.230
3	1.1×10^{16}	58	$+0.041$
4	3.2×10^{16}	48	-0.145
5	9.6×10^{16}	52	$+0.016$
6	2.8×10^{17}	57	$+0.066$
7	8.3×10^{17}	44	-0.130
8	2.5×10^{18}	47	-0.137
9	7.3×10^{18}	33	$+0.035$
mean -0.08; median -0.13			$[-0.25, +0.07]$

Compute cost of loss-based model selection. The ≈2–5× figure quoted in Section 3.2 is obtained by converting a downstream *score* gap into a *compute* multiple, in three steps. *(i) Per-slice score gap.* In each of the nine isoFLOP slices we compare two ways of picking a single run: the *score-optimal* run (argmax composite) and the *loss-optimal* run (argmin normalised BARLOW TWINS loss). Because loss and downstream score are nearly uncorrelated, the loss-optimal run scores lower; averaged over slices the deficit is

$$\overline{\Delta s} \;=\; \tfrac{1}{\#\text{slices}} \sum_{\text{slices}} \big(s_{\text{score-opt}} - s_{\text{loss-opt}}\big) \;=\; 0.0486. \tag{25}$$

(ii) Score–compute slope. Fitting the score-optimal per-slice peak scores against $\log_{10} C$ gives the marginal return of compute, the *score envelope* slope

$$b_s \;=\; \frac{\mathrm{d}\, s}{\mathrm{d}\log_{10} C} \;=\; 0.0886 \quad \text{(score per compute decade)}. \tag{26}$$

(iii) Translate score into compute. Recovering a score deficit $\overline{\Delta s}$ by adding compute requires $\overline{\Delta s}/b_s$ decades, so matching downstream-driven selection by loss-based selection costs a compute multiple

$$10^{\overline{\Delta s}/b_s} \;=\; 10^{0.0486/0.0886} \;=\; 10^{0.548} \;\approx\; 3.5\times, \tag{27}$$

i.e. an excess ("wasted") fraction of $\big(10^{\overline{\Delta s}/b_s} - 1\big) \times 100\% \approx 254\%$. Both inputs ($\overline{\Delta s}$ and b_s) are estimated from nine slice peaks and are sensitive to loss noise, and two effects inflate the point estimate. First, the argmax/argmin choice in step (i) is a winner's-curse estimator: on a subset of configurations (three per bucket, three seeds each) the composite score has a seed standard deviation of about 0.008, and selecting the extreme run over that noise systematically enlarges $\overline{\Delta s}$. Second, $\overline{\Delta s}$ and b_s are each estimated from only nine points; bootstrapping them over the slices puts the excess fraction in a 95% interval of roughly $[156\%,\ 451\%]$, that is, a compute multiple of about 2.6 to $5.5\times$.

H ARCHITECTURE DETAILS

Teacher. The teacher is an all-Transformer dual-branch pixel-wise encoder. Each modality branch (Sentinel-2, 10 optical bands; Sentinel-1, VV/VH with ascending and descending merged) applies a two-layer input MLP ($\text{Linear} \to \text{ReLU} \to \text{Linear}$), a sinusoidal day-of-year positional encoding, and a four-layer Transformer encoder, followed by learned attention pooling. The two per-modality representations are augmented with learned modality embeddings and fused by a two-layer cross-modal Transformer, then reduced to a d_T-dimensional embedding. A batch-normalised MLP projector (4 hidden layers, width 4096; the depth-4, $N_{\text{proj}} = 5P^2$ form of Appendix G) is attached for the BARLOW TWINS/mix-up objective and discarded at inference. The deployed teacher used for all reported results is the $2\,B$ member of a width-scaled family: its per-modality encoders use model width $d_{\text{model}} = 4096$ ($= 4\times$ the latent dimension 1024), four heads, a 16384-wide feed-forward, and a representation dimension $d_T = 1024$. Only the Transformer width scales across the family (Table 6); depth, head count, fusion, pooling, and projector form are fixed. The width is chosen at the compute-optimal point identified in Section 3.

Table 6: **Teacher model family.** Only the Transformer width scales; the $2\,B$ teacher is used for all reported results. The BARLOW TWINS projector ($\sim$84 M, i.e. $5P^2$ at $P = 4096$) is discarded at inference and is width-invariant (Section 3.3), so it is a shrinking fraction of the total as the encoder grows ($14\% \to 8\% \to 4\%$).

	500 M	1 B	2 B (reported)
Latent dim	512	768	1024
Model width d_{model}	2048	3072	4096
Feed-forward width	8192	9216	16384
Representation dim d_T	512	768	1024
Encoder (deployed)	0.52 B	1.01 B	2.06 B
Projector (SSL head, discarded)	84 M	84 M	84 M
Total (training)	0.60 B	1.09 B	2.14 B

Pixel-wise students (N/S/M/L). The students share a single design: two per-modality Transformer backbones (Sentinel-2 and merged Sentinel-1) with sinusoidal DoY encoding and attention pooling, whose outputs are concatenated and passed through an MLP dimensionality reducer (Linear $\rightarrow$ LayerNorm $\rightarrow$ ReLU $\rightarrow$ Dropout $\rightarrow$ Linear) to a 128-dimensional MATRYOSHKA embedding, followed by a final affine-free LayerNorm as in the teacher (Section 4). The four sizes differ only in backbone width and depth (Table 7); the large student is wider rather than deeper. The training-only MATRYOSHKA projection heads add a fixed $\sim$2.35 M parameters (Table 7), independent of student size and discarded at inference, so only the deployed encoder counts at serving time.

Table 7: **Pixel-wise student family.** All four emit a 128-dimensional MATRYOSHKA embedding; only backbone width and depth differ. Encoder parameters are the deployed count (prefix heads discarded at inference).

	Tessera v2-2B-N	Tessera v2-2B-S	Tessera v2-2B-M	Tessera v2-2B-L
Latent dim	36	64	110	160
Backbone width d_{model}	144	256	440	640
Layers per backbone	2	4	4	4
Attention heads	4	4	4	4
Feed-forward width	384	1024	1792	2560
Output dim (MATRYOSHKA)	128	128	128	128
Encoder params (deployed)	1.1 M	7.1 M	21.0 M	43.8 M
Total params (training)	3.4 M	9.5 M	23.4 M	46.2 M

MATRYOSHKA heads. For each prefix $k \in \{16, 32, 64, 128\}$, a small MLP head $h_k : \mathbb{R}^k \rightarrow \mathbb{R}^{d_T}$ maps the student prefix into the teacher embedding space (Equation (2)). The heads are used only during distillation and are discarded at inference; the deployed student outputs the raw 128-dimensional MATRYOSHKA embedding.

I TEACHER PRETRAINING DETAILS

The teacher is pretrained for a single epoch over approximately 14 billion pixel-wise multi-temporal d-pixels with the BARLOW TWINS + mix-up objective of Section 4. Temporal augmentation draws two independent variable-length sub-sequences per d-pixel (Section 4); each modality is additionally dropped out as a whole with a small probability. The training corpus is organised into buckets by valid-observation count, and each step samples across buckets so that a single epoch exposes the model to the full range of temporal coverage rather than to a fixed sequence length. Table 8 lists the optimisation and systems configuration; the learning-rate schedule has no plateau phase (linear warmup directly into cosine decay).

Figure 5 maps the geographic distribution of the pretraining corpus.

Table 8: **Teacher pretraining configuration.**

Setting	Value
Parameters	$2\,B$
Modalities	Sentinel-2 (10 bands); Sentinel-1 (VV/VH, ASC+DESC merged)
Self-supervised objective	BARLOW TWINS ($\lambda_{\text{BT}} = 5 \times 10^{-3}$) + mix-up ($\alpha \sim \text{Beta}(1,1)$, $\lambda_{\text{mix}} = 1.0$)
Teacher embedding dim d_T	1024
Optimiser	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$)
Weight decay	10^{-6}
Peak learning rate	2×10^{-4}
LR schedule	linear warmup (10% of steps) $\rightarrow$ cosine decay to 0
Per-GPU / global batch	256 / 131,072
Epochs	1 (single pass over $\sim$14 B d-pixels)
Mixed precision	AMP, `bf16`
Parallelism	FSDP (hybrid shard: intra-node `FULL_SHARD`, inter-node DDP)
Hardware	512 GPUs (128 $\times$ 4-GPU GB200 NVL72-class nodes)

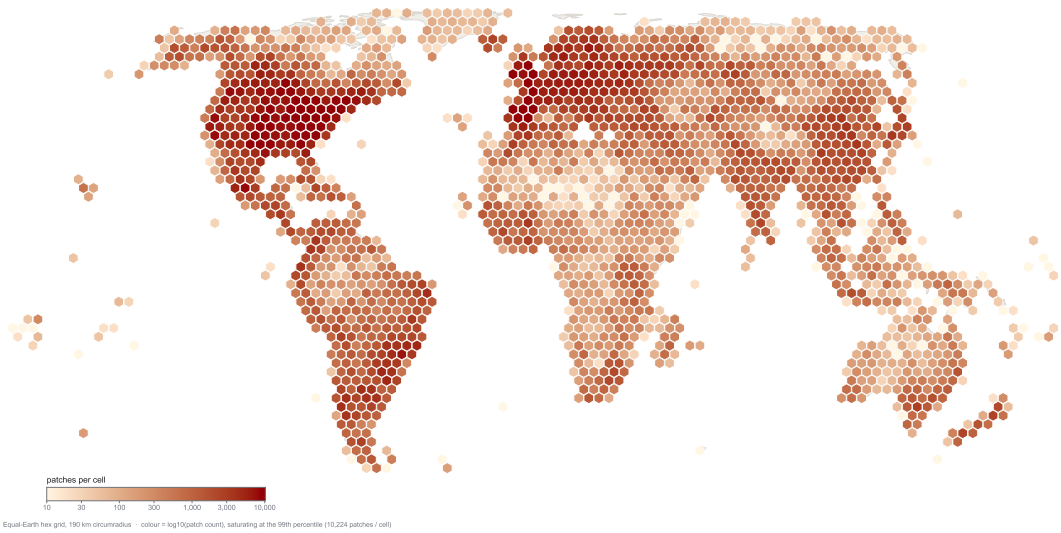

Figure 5: **Geographic distribution of the pretraining corpus.** Each cell of an Equal-Earth hexagonal grid (190 km circumradius) is coloured by the number of 64×64 Sentinel-1/2 patches it contains ($\log_{10}$ scale, saturating at the 99th percentile of 10,224 patches per cell). The corpus comprises 3,420,048 patches; because Tessera v2 is trained pixel-wise, the $\sim$14 billion training d-pixels are obtained by randomly sampling pixels from within these patches.

Mixed precision. Pretraining runs entirely in automatic mixed precision with `bfloat16`: the FSDP `MixedPrecision` policy sets both the parameter dtype and the communication-reduction dtype to `bf16`, and the training loop wraps the forward and loss computation in `torch.autocast(bf16)`. On GB200-class tensor cores `bf16` delivers roughly $1.9\times$ the throughput of `fp32` while leaving the learned representation essentially unchanged (cosine similarity to an `fp32` reference ≈ 1.0), so the speed-up comes at negligible numerical cost.

Systems stack. Pretraining is implemented on a PhysicsNeMo-based training stack (`torch.compile` under FSDP, a streaming data pipeline, topology-matched sharding, and fused-kernel/NCCL/NUMA tuning). These are systems-level optimisations only: the model, loss, optimiser, regularisation, batch, and data are unchanged, so they affect wall-clock throughput but not the learned representation. They make a single epoch over 14 B d-pixels at 512 GPUs practical.

J Distillation Details

The pixel-wise students are distilled against cached teacher embeddings (Section 5) over 1.2 billion pixel-wise d-pixels covering diverse global land cover and phenology; the data-scaling behaviour behind this budget is in Appendix K. Sentinel-2 optical bands are standardised with per-band statistics; Sentinel-1 ascending and descending passes are standardised with their own statistics before being merged.

Per-pixel binning. Valid-observation counts vary widely across pixels. For batch efficiency we bin each pixel's observation count to a fixed ladder of bucket boundaries $\{8, 16, 24, \ldots, 256\}$, rounding up to the nearest bucket and filling the residual slots with a split-and-repeat pattern (pixels exceeding the largest bucket are down-sampled); binning is applied independently to the optical and SAR branches, and pixels sharing the same $(Sentinel-2, Sentinel-1)$ bucket pair are processed in one forward call. This is the distillation-time analogue of the inference bucket sampler of Section 4.

Cross-patch shuffle. To break the spatial correlation between pixels of the same image patch and maximise per-batch statistical diversity, we draw each batch by uniformly sampling pixels from a buffer of many patches, with each pixel sampled exactly once per epoch—the pixel-level analogue of Tessera v1's global d-pixel shuffling.

Distributed implementation. Because each batch yields many variable-length $(Sentinel-2, Sentinel-1)$ bucket groups, each requiring its own forward pass, the per-forward gradient-bucket bookkeeping of standard `DistributedDataParallel` becomes a bottleneck. We instead broadcast rank-0 weights at start-up and, after each local backward pass, flatten all parameter gradients into one contiguous buffer and perform a single all-reduce averaged by world size—equivalent to DDP's averaged gradient but with one collective per step. Training runs as a single flat `while`-loop over global steps with no epoch barriers; all ranks issue an identical collective sequence each step, and on a non-finite loss the step's gradients are zeroed but still all-reduced to keep the collective sequence aligned across ranks. Table 9 lists the configuration.

Table 9: **Distillation configuration.**

Setting	Value
Hardware	$64\times$ H100
Distillation data	1.2 B pixel-wise d-pixels
Teacher targets	cached $d_T = 1024$-d embeddings (fp16)
Student output	128-d MATRYOSHKA ($\{16, 32, 64, 128\}$)
Loss	$\sum_{k\in\{16,32,64,128\}} \big(1 - \cos(h_k(\boldsymbol{s}_{1:k}), \boldsymbol{t})\big)$
Per-GPU batch	8192 pixels
Optimiser	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$)
Peak learning rate	8×10^{-4}
LR schedule	linear warmup (5% of steps) $\rightarrow$ cosine decay to 0
Weight decay	10^{-4}
Gradient clipping	1.0 (global norm)
Dropout	0.1
Mixed precision	`bf16` autocast

K DISTILLATION DATA SCALING

A distilled student inherits its quality from two sources: the teacher it copies, and how efficiently distillation transfers that teacher into the compact student. The first axis, the teacher's own scaling, is characterised in Section 3 and Figure 2f. Here we isolate the second. Holding the teacher fixed at Tessera v2-2B, we vary the data seen during distillation from 0.1 to 1.2 billion d-pixels and track two quantities: how closely the large student reproduces the teacher, measured by the cosine similarity of each MATRYOSHKA prefix to the teacher's d_T=1024-d embedding, and downstream quality on the 15-task ALPHAEARTH suite.

Both curves rise steeply and then flatten (Figure 6, Table 10). Most of the transfer is done within the first 0.6 B d-pixels: past that point every prefix sits within about 0.005 of its final cosine similarity, and the ALPHAEARTH suite composite has already reached 0.589 of its eventual 0.593. The wider prefixes copy the teacher more faithfully at every budget, from 0.952 at d=16 to 0.984 at d=128, as their greater capacity to match the 1024-d target would suggest.

We still distil the deployed student on the full 1.2 B d-pixels. Distillation reads cached teacher embeddings and updates only the student, so it is far cheaper than teacher pretraining (Appendix U); the gain past 0.6 B is marginal but almost free in wall-clock, so we take it. This is the reverse of the pretraining-data trade-off of Section 3, where every extra d-pixel is costly and must be weighed against encoder size.

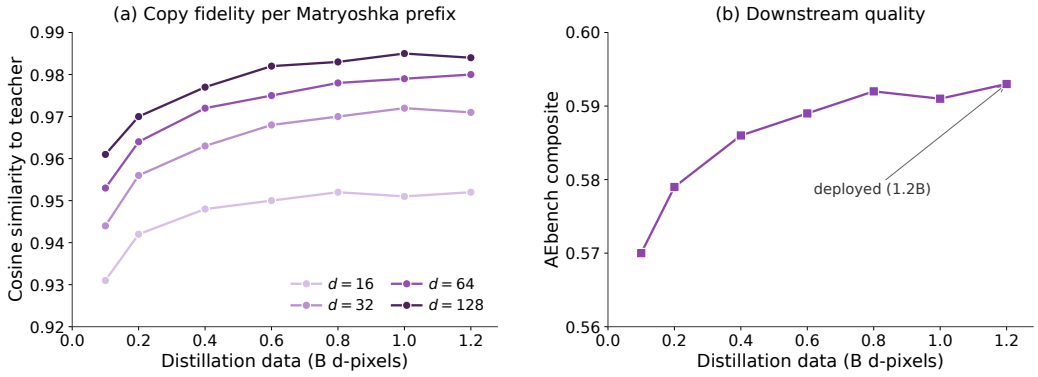

Figure 6: **Distillation data scaling**, teacher fixed at the $2\,B$ Tessera v2 teacher, student Tessera v2-2B-L. **(a)** Cosine similarity between each MATRYOSHKA prefix ($d \in \{16, 32, 64, 128\}$) and the 1024-d teacher embedding. **(b)** ALPHAEARTH suite composite over the 15 tasks. Both saturate near 0.6 B d-pixels; the deployed student uses 1.2 B.

Table 10: **Distillation data scaling** (teacher fixed at the $2\,B$ Tessera v2 teacher; student Tessera v2-2B-L). Cosine similarity of each MATRYOSHKA prefix to the 1024-d teacher and the 15-task ALPHAEARTH suite composite, versus distillation d-pixels. The deployed run (bold) uses 1.2 B.

Distill. data (B)	Cos-16	Cos-32	Cos-64	Cos-128	ALPHAEARTH suite
0.1	0.931	0.944	0.953	0.961	0.570
0.2	0.942	0.956	0.964	0.970	0.579
0.4	0.948	0.963	0.972	0.977	0.586
0.6	0.950	0.968	0.975	0.982	0.589
0.8	0.952	0.970	0.978	0.983	0.592
1.0	0.951	0.972	0.979	0.985	0.591
1.2	**0.952**	**0.971**	**0.980**	**0.984**	**0.593**

L THE 15 ALPHAEARTH SUITE EVALUATION TASKS

The 15 downstream tasks used throughout this paper are drawn from 10 source datasets and cover classification, regression, and change detection across multiple regions and time horizons (Table 11). The same task list underwrites both the downstream-driven scaling-law evaluation (Section 3) and the pixel-wise benchmark (Section 5.1); we do not introduce a separate evaluation suite for the benchmark in order to keep the scaling-law fits and the headline results commensurate.

Table 11: **The 15 AlphaEarth suite evaluation tasks, grouped by source dataset.** Tasks span land cover, land use, change detection, crop typing, biodiversity mapping, and physical-attribute regression. Metrics follow the chance-adjusted convention of Section 3.1 (classification: chance-adjusted balanced accuracy; regression: $\max(0, R^2)$). Parenthesised values are the maximum number of labelled training samples (shots) per task.

Source dataset	Evaluation task(s)	Region / time	Brief description
LCMAP	land cover (300); land use (300); land-cover change (300); land-use change (150)	CONUS, annual	USGS reference data for the contiguous United States; used to evaluate national-scale land-cover and land-use classification, and annual change detection.
LUCAS	land cover (300); land use (300)	Europe, single-date / annual	EU ground-survey points with fine taxonomy and large sample size; tests fine-grained ground-concept discrimination.
GLaNCE	global land cover (300)	Global, annual	Global land-cover training data; tests planetary-scale general-purpose land-cover mapping.
Africa crop mask	cropland / non-cropland (200)	Sub-Saharan Africa, annual	Photo-interpreted cropland reference across multiple Sub-Saharan African countries; tests agricultural-extent classification in regions where existing maps disagree strongly.
Canada / AAFC	coarse crop type (68); fine crop type (75)	Canada, single-date	AAFC in-field crop-type survey; tests hierarchical (coarse-/fine-grained) crop classification at national scale.
Ethiopia / EthCT2020	crop type (49)	Ethiopia, annual	Smallholder field-level crop labels (wheat, barley, maize, teff, ...); tests generalisation to complex smallholder systems with sparse high-quality labels.
US trees (GBIF / iNaturalist)	tree genera, 39 classes (300)	United States, single-date	GBIF / iNaturalist genus-level observations of 39 tree genera; tests biodiversity / species-composition mapping.
Descals oil palm	other / industrial / smallholder oil palm, 3 classes (200)	Global, annual	Global oil-palm plantation reference, three-class; tests fine industrial-vs-smallholder discrimination of a tropical commodity crop.
OpenET ensemble	evapotranspiration, regression (300)	Western US, monthly	Monthly OpenET ET ensemble product; used as a proxy regression task that probes whether the embeddings can approximate a complex multi-model ensemble output.
ASTER GED	surface emissivity, regression (200)	Global, annual	100 m ASTER Global Emissivity Database; tests the representation of a continuous physical surface attribute.

M Held-out Generalisation Suite

The 15 AlphaEarth suite tasks were available throughout model development, so good numbers on them leave open how the embeddings behave on data the design never saw. We therefore assembled a second suite of 14 datasets that we did not consult while building Tessera v2, and ran only the four strongest systems on it. The suite splits into a *vegetation* group (tree species, crop and parcel mapping, forest biomass) and an *urban* group (land use and a range of socioeconomic and environmental layers), and it reaches countries and label types that the AlphaEarth suite tasks under-represent (Table 12).

Why we treat the held-out suite as primary. The two suites are not equally neutral for comparing systems. We used the 15 AlphaEarth suite tasks during development to choose v2 architectures and checkpoints, so our own numbers on them are selected, in part, to look good. Those tasks also come from widely used public label sets, and we cannot assume the baselines avoided them: AlphaEarth does not disclose its training corpus, so its AlphaEarth suite labels may have been seen during its own training. A system that has already seen a task's labels, directly or through a close proxy, holds an advantage that says little about generalisation. None of the 14 held-out datasets were

consulted while building Tessera v2, and they fall outside the label sources the AlphaEarth suite draws on, so we report the held-out composite as the headline number and use the AlphaEarth suite to describe in-distribution behaviour. One caveat: Austrian Crop was a benchmark of the Tessera v1 paper, and v2 inherits v1's regularisers on the strength of those published ablations (Appendix C.6). No v2 architecture, checkpoint, or hyperparameter was selected on it, and v2's margin there is within a seed standard deviation (Table 13), so the held-out lead does not rest on that dataset.

Table 12: **The 14 held-out generalisation datasets**, none used during model development. Metrics are mapped to a common $[0, 1]$ higher-is-better scale before averaging (text below); K is the class count for chance-anchored metrics. All CityRep tasks come from the same multi-city benchmark (Liu et al., 2026).

Dataset	Task (metric)	Region / group	Brief description
TreeSatAI-TS (Astruc et al., 2024)	tree species, multi-label (mAP)	Germany / veg.	Species labels over Sentinel time series; tests fine species discrimination.
PASTIS-R (Sainte Fare Garnot et al., 2022)	parcel segmentation (mIoU, K=18)	France / veg.	Agricultural-parcel labels with radar; tests dense crop-type segmentation.
Austrian Crop (Feng et al., 2026)	crop segmentation (mIoU, K=17)	Austria / veg.	National crop semantic segmentation.
BioMassters (Nascetti et al., 2023)	above-ground biomass (R^2)	Finland / veg.	Forest biomass regression from multi-modal time series.
CropHarvest (Tseng et al., 2021)	crop / non-crop (accuracy, K=2)	Togo; China / veg.	Two regional splits of a global crop benchmark; tests transfer to under-mapped smallholder systems.
CityRep-LUC	land use (macro-F1, K=12)	cities / urban	Zoning / land-use classes (residential, commercial, industrial, green space, ...); tests urban semantic function.
CityRep-RDE	road density (R^2)	cities / urban	OpenStreetMap road network; tests physical street structure.
CityRep-POP	population (R^2)	cities / urban	WorldPop gridded population; tests spatial population intensity.
CityRep-GDP	GDP (R^2)	cities / urban	Gridded GDP product; tests spatial economic output.
CityRep-PM2.5	$PM_{2.5}$ (R^2)	cities / urban	Annual surface $PM_{2.5}$; tests pollution exposure.
CityRep-LST	surface temperature (R^2)	cities / urban	MODIS daytime land-surface temperature; tests surface thermal conditions.
CityRep-NTL	nighttime lights (R^2)	cities / urban	VIIRS annual nighttime-lights product; tests human activity, electrification, and infrastructure intensity.
CityRep-AGE	age distribution ($e^{-\mathrm{KL}}$)	cities / urban	WorldPop age structure; tests demographic composition.

Scoring and the full-suite composite. Each held-out dataset is evaluated at its native metric and mapped to $[0, 1]$ under a higher-is-better convention before any averaging, mirroring the chance-adjusted treatment of the held-in tasks (Section 3.1). Classification and segmentation use the chance-anchored map $(s - 1/K)/(1 - 1/K)$ for K classes (CropHarvest K=2, CityRep-LUC K=12, PASTIS-R K=18, Austrian Crop K=17); metrics already on a natural scale use $\max(0, R^2)$ for the regressions and BioMassters and $\max(0, \mathrm{mAP})$ for TreeSatAI-TS; and the age-distribution match uses $e^{-\mathrm{KL}}$. Averaging the 14 normalised scores with uniform weight gives a held-out composite, which folds into the held-in composite as $c_{29} = (15\, c_{15} + 14\, c_{14})/29$. Every one of the 29 datasets then carries weight $1/29$, so the full-suite number extends the 15-task composite rather than re-weighting it. A system with no prediction for a dataset scores zero there, as in the held-in composite.

Label efficiency. On the held-out datasets we also vary how much supervision the task head receives. For each dataset we train at 1%, 30%, and 100% of the available labels with two heads of

very different capacity, a linear probe and a small <2M-parameter CNN, and read off, per dataset, the gap to the best non-v2 baseline trained the same way (Figure 4f). The Tessera v2 students lead on most datasets at every budget, and the margin is widest at 1%, where the quality of the frozen features matters most and a strong pretrained representation stands in for labels that are not there.

Per-task variability. Table 13 gives the primary metric on each of the six non-CITYREP held-out datasets, with its standard deviation over five seeds. The spread is small on these single-region tasks, at most a few points, so the composite differences of Appendix N are not seed noise. Tessera v2 is best on four of the six, with its widest margins on the dense segmentation tasks (PASTIS-R, Austrian Crop). On BioMassters and CropHarvest-Togo a baseline edges ahead, but in both cases the gap is smaller than a seed standard deviation. The eight CITYREP tasks under the linear probe, with their cross-city deviation, are tabulated separately in Table 15; the CNN-head group composites enter Table 14.

Table 13: **Per-task scores and seed variability on the six non-CITYREP held-out datasets.** Primary metric for the four strongest systems, with the standard deviation over five seeds as a subscript; the best per task is in bold. Metrics are mAP (TreeSatAI-TS), mIoU (PASTIS-R, Austrian Crop), R^2 (BioMassters), and accuracy (CropHarvest); Tessera v2 uses the $d=128$ student. These are the raw vegetation-group scores behind the held-out composite (Appendix N).

Model	TreeSatAI mAP↑	PASTIS-R mIoU↑	Austrian mIoU↑	BioMass. R^2↑	CH-Togo acc↑	CH-China acc↑
Tessera v2-2B-L	**0.832** $_{0.018}$	**0.666** $_{0.021}$	**0.461** $_{0.031}$	0.813 $_{0.019}$	0.767 $_{0.020}$	**0.805** $_{0.022}$
Tessera v1	0.816 $_{0.011}$	0.536 $_{0.023}$	0.455 $_{0.027}$	0.801 $_{0.009}$	**0.780** $_{0.014}$	0.722 $_{0.008}$
ALPHAEARTH	0.783 $_{0.028}$	0.542 $_{0.025}$	0.236 $_{0.007}$	**0.816** $_{0.033}$	0.774 $_{0.032}$	0.785 $_{0.024}$
OlmoEarth-L	0.798 $_{0.010}$	0.512 $_{0.006}$	0.385 $_{0.013}$	0.804 $_{0.029}$	0.778 $_{0.015}$	0.725 $_{0.012}$

N ROBUSTNESS OF THE HELD-OUT RANKING

Tessera v1 embeddings have already been validated by the community across a broad range of vegetation tasks, including tree-species mapping, crop-type classification, and biomass and canopy-height regression (Feng et al.; Ball et al., 2026; Lisaius et al., 2026). This body of work establishes that the BARLOW TWINS-trained Tessera family carries strong predictive signal for vegetation, so a held-out suite drawn primarily from that domain would largely re-confirm what is already known. Urban and socioeconomic settings, by contrast, remain comparatively under-evaluated. We therefore oversampled them when assembling the held-out suite, which is why CITYREP (Liu et al., 2026) supplies eight of the 14 tasks.

This deliberate imbalance concentrates weight on a single benchmark. Several of the CITYREP targets, among them nighttime lights, GDP, population, and land-surface temperature, are spatially correlated, so a uniform mean over the fourteen tasks assigns one benchmark family more than half of the effective weight. It is therefore fair to ask whether the held-out headline depends on that family. We test the ranking in three ways: by removing CITYREP entirely, by replacing the scoring convention with two alternatives, and by rescoring the CITYREP tasks with the original benchmark's point linear probe in place of the CNN head. Tessera v2-2B-L finishes first in every case (Table 14).

Removing CITYREP does not change who leads. On the six vegetation tasks alone, Tessera v2-2B-L scores 0.644, and its margin over the strongest baseline grows rather than shrinks, from $+0.033$ across all fourteen tasks to $+0.052$ once the urban family is gone. On the eight CITYREP tasks by themselves it is first again, now only narrowly ahead of OlmoEarth-L. The one thing that moves between the two slices is the runner-up: Tessera v1 trails closest on vegetation, OlmoEarth-L on the urban tasks.

Scoring CITYREP by the original linear probe is the most adversarial of the three tests for Tessera v2, whose urban edge comes largely from a small CNN over the 64×64 patch around each point (Appendix P). Replacing that CNN with the benchmark's point probe on all eight CITYREP tasks, and folding the result back into the fourteen-task composite, OlmoEarth-L does move ahead on the CITYREP subset (0.574 versus 0.562), in line with its stronger point-feature readout. Over the full fourteen tasks, however, Tessera v2 still leads (0.597 versus 0.573), because its margin on the six

Table 14: **The held-out ranking survives every stress test.** Top block: the chance-adjusted composite (the headline scheme) over different task subsets, and over all 14 tasks with the CITYREP scores taken from the original point linear probe rather than the CNN head. Bottom block: all 14 tasks under two alternative aggregations. Tessera v2-2B-L is first in every row (bold).

	Tessera v2-L	OlmoEarth-L	Tessera v1	ALPHAEARTH
Chance-adjusted composite (headline scheme)				
All 14 tasks	**0.647**	0.614	0.614	0.590
CITYREP only (8)	**0.649**	0.644	0.630	0.604
Excluding CITYREP (6)	**0.644**	0.573	0.592	0.570
All 14, CITYREP linear probe	**0.597**	0.573	0.560	0.557
All 14 tasks, alternative aggregations				
Plain mean of raw metrics	**0.685**	0.658	0.658	0.631
Per-task min-max, then mean	**0.790**	0.596	0.433	0.318

vegetation tasks (0.644 versus 0.573) outweighs the urban gap. Even under the evaluation head that most favours the baselines, the held-out headline does not change hands.

The averaging rule matters just as little. The headline maps each metric to $[0, 1]$ with a chance-adjustment and averages with uniform weight, but that is one reasonable recipe among several. Averaging the raw native metrics with no chance-adjustment keeps Tessera v2-2B-L first (0.685), and so does a per-task min-max normalisation across the four systems (0.790). Min-max rewards being best on each task rather than best on average, and it widens the gap here because Tessera v2-2B-L is top or near-top on most individual tasks. Neither the CITYREP family nor the specific averaging rule carries the held-out result.

O EVALUATION PROTOCOL AND DATA PREPARATION

Baselines. Section 5.1 names only the systems that carry the headline comparison. The full ALPHAEARTH suite comparison set comprises the two directly comparable embedding-as-data systems, ALPHAEARTH (Brown et al., 2025) and Tessera v1 (Feng et al., 2026), together with PRESTO (Tseng et al., 2024), OLMOEARTH (Herzog et al., 2026), MOSAIKS (Rolf et al., 2021), and a suite of remote-sensing foundation models and generic backbones (Fuller et al., 2023; Guo et al., 2024; Reed et al., 2023; Tseng et al., 2025; Astruc et al., 2025; Dosovitskiy et al., 2021; Ronneberger et al., 2015; Klemmer et al., 2025a; Bastani et al., 2023; Liu et al., 2024; Xiong et al., 2024; Jakubik et al., 2023; Hong et al., 2024; Wu et al., 2025b; Clay Foundation, 2024; Forgaard et al., 2026; Jakubik et al., 2025). The four strongest on this suite (Tessera v2, Tessera v1, ALPHAEARTH, and OLMOEARTH) carry forward to the held-out suite.

Every system is scored on the shared ALPHAEARTH suite (Appendix L) and on the held-out suite (Appendix M) through a single procedure. Where ALPHAEARTH (Brown et al., 2025) already reports a baseline on the shared suite we adopt its published number, since re-running the model under an identical setup would only reproduce it; the remaining systems we run ourselves, staying as close to the ALPHAEARTH evaluation as their released weights allow so that every figure is obtained the same way.

The shared-suite annotations are sparse geographic points rather than dense masks, which leaves open how a spatially structured model should turn a point into a feature vector. For our own students and the other embedding-as-data systems the question does not arise: they emit a per-pixel 10 m embedding, so we read the embedding at the labelled pixel directly. It does arise for backbones such as ViT (Dosovitskiy et al., 2021), Prithvi, and Clay (Clay Foundation, 2024), which return only a coarse grid of tokens, and following ALPHAEARTH we resolve it by reading a single feature at each label instead of propagating the label across the chip or training a dense decoder on the backbone. Writing the encoder output as a feature grid $F \in \mathbb{R}^{H_t \times W_t \times D}$ coarser than the $H \times W$ input, we regard each token as a georeferenced cell and place the label, at its exact coordinate, at a fractional grid position (u, v). The point feature is the bilinear interpolation of the four enclosing tokens $\mathcal{N}(u, v)$,

$$e(p) = \sum_{(a,b)\in\mathcal{N}(u,v)} w_{ab}\, F_{a,b}, \qquad w_{ab} = \big(1 - |u - a|\big)\big(1 - |v - b|\big), \tag{28}$$

with $\sum_{(a,b)} w_{ab} = 1$. This coincides with resampling every channel of F to 10 m and sampling at the label, but reads the fractional coordinate directly, so it needs neither an upsampling decoder nor any rounding of the label to the nearest token. When the encoder returns one grid per acquisition we concatenate the per-date features into $e(p) \in \mathbb{R}^{TD}$, and each labelled point i enters the evaluation as a single pair (e_i, ℓ_i), scored, following AlphaEarth, by 1- and 3-nearest-neighbour classifiers and a linear probe, of which we report the best.

Reducing a patch to one point in this way is deliberate rather than a convenience. Because each chip is cropped so that its target lies at the centre, a *pointwise* (1×1) linear decoder trained on the patch with only that centre pixel labelled receives gradient at a single position and converges to the same head the centre probe fits in closed form, so for such a decoder extracting the interpolated centre feature is the identical experiment without the decoder. This equivalence is specific to a pointwise readout: a spatial decoder that aggregates neighbouring context, such as UPerNet, is not reducible to the centre probe. We probe the pointwise representation on purpose, because what it isolates is the quality of a model's per-pixel representation—what the shared suite is meant to probe—rather than the capacity of a segmentation head or the artefact of copying one label across an entire patch.

How these scores should be read depends on how coarse the underlying grid is, and Prithvi 1.0 (Jakubik et al., 2023) makes the dependence concrete. It ingests 224×224 HLS L30 chips at 30 m, a window of about 6.72 km, over up to three acquisitions; with a patch size of 16 it emits a 14×14 grid of 768-dimensional tokens, so one token summarises roughly 480 m of ground and the concatenated point feature has dimension $768\,T \in \{768, 1536, 2304\}$. Interpolating this field to 10 m yields a visually smooth raster but no additional spatial information, and the effective support of the extracted feature stays at the ≈480 m token. We keep this in mind when comparing a low-resolution backbone against natively high-resolution embeddings.

All imagery is retrieved from the Microsoft Planetary Computer. We take Sentinel-2 from the L2A collection[2] and Sentinel-1 from either the RTC[3] or the GRD[4] product, according to which one the model in question was trained on. Each modality is normalised with the statistics published for that model, and its Sentinel-2 cloud and haze handling is reproduced wherever it is documented, but we apply no further radiometric processing: corrections that some pipelines perform, such as the σ^0 conversion and thermal-noise removal that THOR (Forgaard et al., 2026) applies to Sentinel-1 GRD, are omitted so that a score reflects the released model on standard inputs rather than our own preparation.

The held-out suite follows the same retrieval, normalisation, and readout. The only labels that need separate handling are those of CityRep (Liu et al., 2026), which attach to individual city locations rather than to image tiles; for each we crop a 64×64 patch centred on the labelled point. The linear and nearest-neighbour probes read its centre feature as above, while the small CNN head takes the whole patch and can use the surrounding spatial context; Appendix P contrasts the two heads on CityRep.

P The Evaluation Head Matters: CityRep under a Linear Probe

The held-out CityRep numbers in the main text use a small CNN over each label's 64×64 point-centred patch, which can exploit local spatial structure. The original CityRep benchmark (Liu et al., 2026) instead reads each representation with a linear probe on the point feature alone. To compare like for like against those published numbers, we rerun the same linear probe on all four systems: Tessera v1 and AlphaEarth are the values reported by Liu et al. (2026), while Tessera v2-2B-L and OlmoEarth-L are evaluated here (Table 15).

Under the linear probe, OlmoEarth-L and Tessera v2-2B-L are the two strongest systems and sit close together, with OlmoEarth-L just ahead (mean rank 1.50 versus 1.75) and both well clear of AlphaEarth and Tessera v1 (3.25 and 3.50). Their strengths split by task type: OlmoEarth-L leads on the socioeconomic and land-use layers (population, GDP, nighttime lights, $PM_{2.5}$, land use, and demographics), whereas Tessera v2 is best on the more physically structured targets, road density and surface temperature, and never places below second; OlmoEarth-L is the weakest of the four on road

[2] https://planetarycomputer.microsoft.com/dataset/sentinel-2-l2a
[3] https://planetarycomputer.microsoft.com/dataset/sentinel-1-rtc
[4] https://planetarycomputer.microsoft.com/dataset/sentinel-1-grd

density. Switching to a small CNN head tips this near-tie the other way: on the same CITYREP tasks Tessera v2 takes the best composite (Appendix N), edging past OlmoEarth-L. The lever is the head. A linear probe reads only the centre feature and discards the surrounding 64×64 patch, whereas a small CNN over that patch can use the local spatial structure that Tessera v2's pixel-wise embeddings carry. A dataset's headline number therefore reflects the evaluation head as much as the embedding, and pixel-wise representations like Tessera v2 are best read with a head that can use their spatial neighbourhood.

Table 15: **Linear-probe results on CITYREP.** Following the original benchmark (Liu et al., 2026), each representation is read by a linear probe on the point feature, with no spatial head. Per task we report the mean primary metric across the 8 cities and 5 seeds, with the cross-city standard deviation as a subscript; the best per task is in bold. Tessera v1 and ALPHAEARTH are the published numbers (Liu et al., 2026); Tessera v2-2B-L and OlmoEarth-L are evaluated here. Metrics are macro-F1 for LUC, symmetric KL for AGE (lower is better), and R^2 otherwise; the final column is the mean task-level rank over the four systems (lower is better). Contrast the CNN-head ordering of Appendix N.

Model	LUC F1↑	RDE R^2↑	POP R^2↑	AGE KL↓	GDP R^2↑	NTL R^2↑	$PM_{2.5}$ R^2↑	LST R^2↑	Rank ↓
Tessera v2-2B-L	$0.264_{0.059}$	$\mathbf{0.551}_{0.074}$	$0.624_{0.126}$	$0.040_{0.017}$	$0.611_{0.121}$	$0.641_{0.129}$	$0.457_{0.145}$	$\mathbf{0.456}_{0.148}$	1.75
Tessera v1	$0.258_{0.067}$	$0.538_{0.086}$	$0.592_{0.140}$	$0.043_{0.020}$	$0.596_{0.134}$	$0.595_{0.143}$	$0.445_{0.159}$	$0.373_{0.161}$	3.50
ALPHAEARTH	$0.254_{0.076}$	$0.514_{0.087}$	$0.615_{0.158}$	$0.042_{0.019}$	$0.598_{0.127}$	$0.632_{0.105}$	$0.436_{0.149}$	$0.442_{0.139}$	3.25
OlmoEarth-L	$\mathbf{0.268}_{0.091}$	$0.503_{0.104}$	$\mathbf{0.658}_{0.174}$	$\mathbf{0.038}_{0.026}$	$\mathbf{0.662}_{0.152}$	$\mathbf{0.667}_{0.148}$	$\mathbf{0.488}_{0.183}$	$0.447_{0.172}$	1.50

Q FULL PIXEL-WISE RESULTS

Table 16 gives the per-dataset FIRST-d scores of the distilled pixel-wise student behind Figure 8, at the maximum shot per dataset, for $d \in \{16, 32, 64, 128\}$, alongside the naive MATRYOSHKA-BARLOW TWINS baseline. This student is a preliminary distillation checkpoint used only for the mechanism study of Appendix V, not one of the deployed N/S/M/L models; its absolute composite (0.517 at d=128) therefore sits well below the headline Tessera v2-2B-L (0.593), and only the distilled-versus-naive gap is meaningful here. OpenET is evaluated at the monthly inference window (Table 2), which is why its entries exceed the annual-window score of the deployed model; all other tasks use the annual window. Distilled MATRYOSHKA dominates the naive baseline at every reported prefix dimension; the gap is largest at d=16 and narrows as d grows, consistent with the interpretation (Appendix V) that the naive method receives stronger gradient pressure on early coordinates but does not learn a robust semantic ordering.

Table 16: **Per-dataset MATRYOSHKA prefix scores** for the pixel-wise distillation student of Figure 8 (FIRST-d, maximum shot). Top: distilled MATRYOSHKA; bottom: naive MATRYOSHKA-BARLOW TWINS. Per-dataset entries are raw balanced accuracy (classification / change detection) and R^2 (regression); OpenET is scored at the monthly inference window of Table 2, all other tasks at the annual window; the **Composite** row is the chance-adjusted downstream score of Section 3.1.

	Distilled MATRYOSHKA				Naive MATRYOSHKA-BARLOW TWINS			
Dataset	16	32	64	128	16	32	64	128
Africa crop mask	0.840	0.867	0.852	0.854	0.800	0.835	0.838	0.849
ASTER GED	0.770	0.788	0.805	0.819	0.735	0.770	0.793	0.810
CA crops (coarse)	0.460	0.521	0.547	0.555	0.447	0.504	0.530	0.545
CA crops (fine)	0.407	0.465	0.489	0.509	0.386	0.443	0.476	0.499
Descals oil palm	0.626	0.703	0.690	0.690	0.604	0.680	0.685	0.681
Ethiopia crops	0.349	0.354	0.376	0.378	0.348	0.348	0.376	0.370
GLanCE	0.616	0.625	0.645	0.664	0.562	0.603	0.632	0.642
LCMAP-LC	0.749	0.773	0.793	0.800	0.714	0.745	0.778	0.791
LCMAP-ΔLC	0.740	0.762	0.756	0.747	0.716	0.738	0.736	0.745
LCMAP-LU	0.746	0.771	0.798	0.804	0.699	0.747	0.781	0.791
LCMAP-ΔLU	0.757	0.748	0.742	0.731	0.708	0.722	0.731	0.722
LUCAS-LC	0.306	0.345	0.384	0.407	0.285	0.319	0.367	0.383
LUCAS-LU	0.318	0.340	0.361	0.371	0.293	0.318	0.345	0.350
OpenET ET	0.477	0.564	0.617	0.636	0.440	0.532	0.601	0.619
US trees	0.143	0.157	0.177	0.184	0.123	0.140	0.166	0.171
Composite	**0.454**	**0.493**	**0.510**	**0.517**	**0.413**	**0.463**	**0.492**	**0.503**

R MATRYOSHKA FAILURE ANALYSIS

Coordinate-exchangeability seed variance. The RANDOM-16 scores of Table 23 are averaged over five seeds. Across the classification subset the seed standard deviation is small—0.001–0.013 for distilled MATRYOSHKA and 0.004–0.016 for naive MATRYOSHKA-BARLOW TWINS (Table 17)—and well below the 0.044 FIRST-VS-RANDOM gap of the naive model, so the observed ordering is not a seed-noise artefact.

Table 17: **RANDOM-16 seed standard deviation** across five seeds, per classification dataset.

Dataset	Distilled σ	Naive σ
Africa crop mask	0.008	0.016
GLaNCE	0.008	0.010
LCMAP-LC	0.007	0.004
LCMAP-LU	0.013	0.013
Descals oil palm	0.006	0.004
LUCAS-LC	0.009	0.010
US trees	0.001	0.005

Per-coordinate scale staircase. Both models exhibit a staircase in per-coordinate standard deviation σ_d aligned with the prefix multiplicities m_j of Equation (33) (Table 18). For the distilled model this is expected: the distillation objective acts on the same nested prefixes and therefore carries the same multiplicity $4/3/2/1$ by construction. What separates the two is scale and shape. The naive model's σ_d is 1.6–1.8$\times$ larger in every group, its early coordinates ($m_j = 4$) are heavy-tailed and bimodal, and its tail does not return to the single-prefix scale, whereas the distilled step is a scale factor on otherwise Gaussian-like marginals (Figure 8a). This is the quantitative form of Figure 8(c).

Table 18: **Per-coordinate scale by prefix-loss multiplicity.** Mean per-dimension standard deviation σ_d within each coordinate group, measured for both models at the same point—on the representation *before* the final affine-free LayerNorm. (The LayerNorm renormalises every deployed embedding to $\sum_d \sigma_d^2 = 128$, so the coordinate-wise staircase is only visible pre-normalisation; reading both models there keeps the comparison like-for-like.) Both models step with m_j; the naive model's inflation is 1.6–1.8$\times$ larger in every group and is accompanied by non-Gaussian marginals, the gauge pathology of Observation 1.

Group	m_j	Distilled σ_d	Naive σ_d	Naive rel. grad. norm
1–16	4	$\sim$1.15	$\sim$1.85	3.92
17–32	3	$\sim$0.85	$\sim$1.5	2.93
33–64	2	$\sim$0.80	$\sim$1.3	1.97
65–128	1	$\sim$0.75	$\sim$1.2	1.00

Gradient norm per coordinate group. The last column of Table 18 reports the naive model's mean gradient norm per coordinate group, normalised to the $m_j = 1$ tail. It tracks the prefix-loss multiplicity almost exactly ($\approx m_j$-fold). The distillation objective has the same multiplicity by construction, which is why the distilled model shows a σ_d step as well; gradient counting alone therefore cannot produce a semantic coordinate order. What differs is what the extra gradient carries: in the naive model it comes from decoupled heads whose constraints conflict (Appendix V), yielding the inflated, non-Gaussian early coordinates, whereas in the distilled model every prefix gradient points at the same fixed teacher target. This is why the naive FIRST-d advantage over RANDOM-d is an optimisation artefact, while the distilled ordering survives the coordinate-exchangeability test (Appendix W.2).

Effective rank, CKA, and Procrustes alignment. Student prefixes retain higher normalised effective rank and stronger alignment to the full teacher embedding than teacher coordinate-prefixes at the same k (Table 19); the gap is largest at $k = 16$ and narrows steadily with k (the teacher is d_T-dimensional, so even $\boldsymbol{t}_{1:128}$ is a strict prefix). We report *normalised effective rank*—the effective rank (the entropy of the normalised singular-value spectrum) divided by the prefix dimension k—so the metric lies in $[0, 1]$ and measures the fraction of the k available axes that the prefix actually

exercises. The student prefix uses ~ 0.9 of its axes at every k, whereas the teacher coordinate-prefix uses barely half at $k = 16$.

Table 19: **Spectral and alignment diagnostics** for student prefix $\boldsymbol{s}_{1:k}$ vs. teacher coordinate-prefix $\boldsymbol{t}_{1:k}$, against the full teacher embedding. Normalised effective rank $\in [0, 1]$ is the effective rank divided by k.

	Norm. eff. rank		CKA to $\boldsymbol{t}$		Procrustes resid.	
k	student	teacher	student	teacher	student	teacher
16	0.92	0.54	0.742	0.493	0.412	0.611
32	0.91	0.58	0.831	0.621	0.323	0.512
64	0.92	0.65	0.912	0.764	0.218	0.392

Teacher-target dimensionality. The gradient-multiplicity argument concerns the within-128 coordinate gauge that BARLOW TWINS does not fix, and is independent of the teacher target dimension d_T: changing d_T changes the distillation compression ratio (d_T/d, from $8\times$ at d=128 to $64\times$ at d=16) but does not remove the unequal self-supervised gradient pressure in naive prefix-BT. Empirically, a $d_T = 512$ target gives slightly lower but monotonically identical prefix retention to the production $d_T = 1024$ target: the $d = 16/32/64/128$ composites are $0.441/0.485/0.498/0.506$ at $d_T = 512$ versus $0.454/0.493/0.510/0.517$ at $d_T = 1024$.

S SCALING STUDY: ADDITIONAL MATERIAL

S.1 SCOPE OF THE FINDINGS

F1 and **F2** are properties of the architectural family we sweep: pixel-wise Sentinel-1/2 encoders pretrained with BARLOW TWINS (Zbontar et al., 2021) at short sequence lengths $L \in \{8, 16\}$. They are not advanced as universal EO scaling laws. We do not claim, and our experiments do not establish, that: (a) patch-level architectures with masked-image-modelling objectives obey the same exponents; (b) the same exponents extend to other satellite modalities (e.g., hyperspectral or sub-meter optical imagery); (c) the projector-saturation pattern would replicate under contrastive or generative pretraining objectives whose projector plays a different role; (d) the compute-waste quantification associated with loss-based model selection is invariant to evaluation suite—it is computed against the 15-task composite of this study, and a different task mixture would give a different number.

S.2 WHY THE LOSS CORRELATES SO WEAKLY WITH DOWNSTREAM SCORE

We attribute the weak loss–score correlation of **F1** to four properties of EO that are not jointly present in the language and natural-image regimes where loss-based scaling laws have been validated: (i) heterogeneous downstream tasks with different sensitivities to the representation; (ii) cloud and orbital sampling whose variance dominates the loss without being downstream-relevant; (iii) multi-sensor signals with different physical meaning and noise characteristics; and (iv) the susceptibility of redundancy-reduction objectives to shortcut invariances orthogonal to the physical processes downstream tasks depend on.

S.3 LOSS-OPTIMAL VERSUS SCORE-OPTIMAL SELECTION

Within each of the nine compute buckets of Figure 2b–d we contrast two selection criteria: the run with the highest downstream score (*score-optimal*) and the run with the lowest normalised BARLOW TWINS loss (*loss-optimal*). Fitting power laws through the resulting bucket peaks separately gives $N^{\star}_{\text{enc}} \propto C^{0.36}$ for score-optimal selection but a nearly flat $N^{(\ell)}_{\text{enc}} \propto C^{0.02}$ for loss-optimal selection; the two laws share roughly the same low-compute behaviour but diverge as C grows. Translating the per-bucket score gap (mean gap 0.049 along a score-envelope slope of 0.089 per compute decade) into compute units yields the 2–5$\times$ figure quoted in the main text (derivation in Appendix G, Equation (27)). For every run we record the converged BARLOW TWINS loss on a held-out d-pixel validation set, normalised by the projector-determined ceiling $P + \lambda_{\text{BT}} P(P-1)$ so that runs with different projector widths are comparable.

S.4 FITTED EXPONENTS

Table 20 reports the fitted exponents with bootstrap confidence intervals and leave-one-bucket-out (LOBO) stability.

Table 20: Fitted scaling exponents (395 runs; bootstrap 95% CI; LOBO mean$\pm\sigma$ over nine leave-one-out fits, seven for the projector).

Component	Exponent	95% CI	LOBO mean$\pm\sigma$
Encoder $N^{\star}_{\mathrm{enc}}(C)$	0.36	$[+0.31, +0.39]$	0.36 ± 0.01
Projector $N^{\star}_{\mathrm{proj}}(C)$	0.011	$[-0.02, +0.04]$	0.01 ± 0.01
Data $D^{\star}(C)$	0.63	$[+0.61, +0.69]$	0.63 ± 0.01
Loss-optimal encoder $N^{(\ell)}_{\mathrm{enc}}(C)$	$\propto C^{0.02}$		

T TEACHER OBJECTIVE DETAILS

The two views Y_A, Y_B of each d-pixel are encoded by the dual-branch encoder and projected to $\mathbb{R}^P$, giving batch-normalised embeddings $Z_A, Z_B \in \mathbb{R}^{B\times P}$. The BARLOW TWINS loss enforces invariance to the random temporal sampling while decorrelating embedding dimensions:

$$\mathcal{L}_{\mathrm{BT}} \;=\; \sum_i \big(1 - C_{ii}\big)^2 + \lambda_{\mathrm{BT}} \sum_i \sum_{j\neq i} C_{ij}^2, \qquad C \;=\; \tfrac{1}{B}\, Z_A^\top Z_B, \tag{29}$$

with $\lambda_{\mathrm{BT}} = 5\times 10^{-3}$ following Zbontar et al. (2021).

The mix-up regulariser enforces a linear-consistency property: linear interpolation in the input space should correspond to linear interpolation in the embedding's cross-correlation space. At each step we sample $\alpha_{\mathrm{mix}} \sim \mathcal{U}(0,1)$, construct a batch-shuffled view $Y_S = \mathrm{Shuffle}(Y_B)$, mix $Y_M = \alpha_{\mathrm{mix}} Y_A + (1-\alpha_{\mathrm{mix}}) Y_S$, and compare the mixed embedding's cross-correlations against the interpolation-predicted targets:

$$\mathcal{L}_{\mathrm{MIX}} \;=\; \tfrac{1}{2}\big\|C^{MA} - C^{MA}_{\mathrm{target}}\big\|_F^2 + \tfrac{1}{2}\big\|C^{MS} - C^{MS}_{\mathrm{target}}\big\|_F^2, \tag{30}$$

where $C^{MA} = \frac{1}{B} Z_M^\top Z_A$, $C^{MS} = \frac{1}{B} Z_M^\top Z_S$ (normalised by B exactly as C in Equation (29), so λ_{mix} does not depend on batch size), and the targets are the convex combinations $\frac{1}{B}\big(\alpha_{\mathrm{mix}} Z_A^\top Z_\cdot + (1-\alpha_{\mathrm{mix}}) Z_S^\top Z_\cdot\big)$ predicted by linear interpolation; the full derivation is in Appendix C. The total objective is

$$\mathcal{L}_{\mathrm{total}} \;=\; \mathcal{L}_{\mathrm{BT}} + \lambda_{\mathrm{mix}}\, \mathcal{L}_{\mathrm{MIX}}, \qquad \lambda_{\mathrm{mix}} = 1.0, \tag{31}$$

with λ_{mix} recommended by Bandara et al. (2023). We retain Tessera v1's global shuffling of d-pixels across MGRS tiles to prevent spatial-autocorrelation bias during batching; Tessera v1 ablations show that removing either mix-up or global shuffling degrades validation accuracy substantially under temporal sparsity.

U DEPLOYMENT COST AND CHOOSING AN ARTEFACT

Figure 7 summarises the inference cost of one global, annual, 10 m Sentinel-1/2 pass, expressed in single-H100-year equivalents (GPUy). Across the encoder-size range spanned by our students and the teacher, the cost follows a near-linear power law in the encoder parameter count N (in millions),

$$\mathrm{GPUy}(N) \;\approx\; 0.041 \cdot N^{1.03}, \tag{32}$$

fit to the inference-FLOPs accounting of Appendix B.8 and the per-FLOP throughput of the H100 accelerator. Over our students and teacher this gives $\approx$100 GPUy for the 2 B teacher and $\approx$0.3, 0.9, and 2 GPUy for Tessera v2-2B-S (7 M), -M (21 M), and -L (44 M) respectively.

Encoder inference dominates wall-clock cost. Equation (32) counts encoder forward-pass FLOPs, which is the quantity that matters: in the full embedding-as-data pipeline (Figure 7b), the encoder forward pass, split almost evenly between the Sentinel-2 and Sentinel-1 backbones ($\sim 47\%$ each), accounts for $\sim 94\%$ of wall-clock time, while data preparation adds only $\sim 6\%$ (temporal sampling and stacking 2.4%, disk I/O for writing embeddings 2.3%, and normalisation/fusion/quantisation 1.3%). The full pipeline is therefore only $\sim 1.06\times$ the encoder-only cost, so the GPUy estimates of Figure 7(a) are a tight proxy for end-to-end cost.

Choosing an artefact in practice. Different users have different binding constraints; Figure 7(c) summarises which artefact and prefix dimension we recommend for each regime. On-device and edge deployments (field apps, mobile, real-time) are best served by Tessera v2-2B-S at $d \in \{16, 32\}$; storage-constrained regional pipelines by Tessera v2-2B-S/M at $d \in \{32, 64\}$; balanced provider-side products by Tessera v2-2B-M/L at $d \in \{64, 128\}$; and research or further-supervision use cases by the $2\,B$ teacher at $d = 128$.

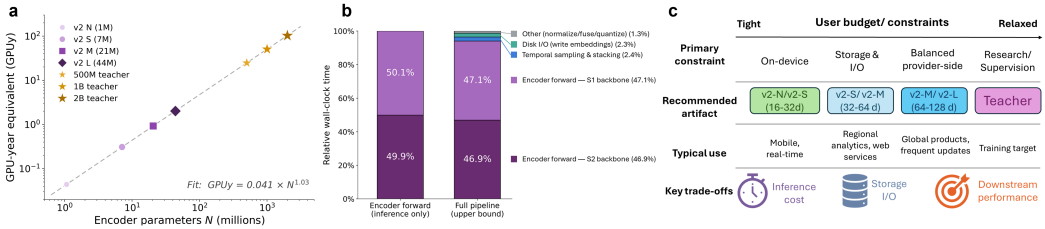

Figure 7: Deployment cost of the student family. **(a)** GPUy cost of one annual global pass vs. encoder size, with the fit of Equation (32); markers are the Tessera v2 students and the 500M/1B/2B teachers. **(b)** Wall-clock breakdown of the embedding-as-data pipeline: the encoder forward pass accounts for $\sim 94\%$. **(c)** Recommended artefact and prefix dimension by user-budget regime.

V WHY NAIVE MATRYOSHKA SELF-SUPERVISION FAILS

A natural baseline is to add MATRYOSHKA prefixes during self-supervised pretraining: have a single encoder produce $\boldsymbol{z} \in \mathbb{R}^{128}$, attach a separate projector g_k to each prefix $\boldsymbol{z}_{1:k}$, and minimise $\sum_{k \in \mathcal{K}} \mathcal{L}_{\mathrm{BT}}\big(g_k(\boldsymbol{z}^A_{1:k}), g_k(\boldsymbol{z}^B_{1:k})\big)$. We tried this and it does not work (Appendix W.2). Below we offer a hypothesis for the failure, stated as a conjecture rather than a proof; if correct, it generalises beyond BARLOW TWINS and beyond EO.

Rotation symmetry. BARLOW TWINS identifies representation *subspaces* only up to orthogonal transformations: if $Z \in \mathbb{R}^{B \times 128}$ is a feasible (decorrelated, invariant) solution then so is ZQ for any orthogonal $Q \in \mathbb{R}^{128 \times 128}$. Coordinate ordering within the embedding is therefore a gauge degree of freedom; nothing in the loss specifies which axes should carry which information.

Prefix-BT breaks the symmetry through gradient multiplicity, not through a semantic signal. Let $\mathcal{K} = \{16, 32, 64, 128\}$ be the prefix set, and define the *gradient multiplicity* of coordinate j as the number of prefix losses that act on it,

$$m_j \;=\; \big|\{k \in \mathcal{K} : j \le k\}\big| \;=\; \begin{cases} 4 & j \in [1, 16], \\ 3 & j \in [17, 32], \\ 2 & j \in [33, 64], \\ 1 & j \in [65, 128]. \end{cases} \tag{33}$$

The total gradient feeding coordinate j is the sum of m_j contributions from prefix losses $\{\mathcal{L}^{(k)}_{\mathrm{BT}}\}_{k \ge j}$. Adding these losses does break the rotation symmetry of single-prefix BARLOW TWINS, but it breaks it through gradient pressure: nothing in the construction specifies that the most informative axes should be the early ones, only that early ones receive more gradient.

An implicit Gaussian baseline that prefix-BT disturbs. The visible footprint of this gradient imbalance is best understood against a baseline that the loss itself does not impose. BARLOW TWINS

constrains only the cross-correlation matrix of the projector outputs to be diagonal, and imposes no marginal-distribution constraint. Empirically, however, single-prefix BARLOW TWINS reliably produces approximately Gaussian per-coordinate marginals. Three mechanisms cooperate to produce this implicit bias: (i) batch normalisation on the projector output fixes the first two moments; (ii) each projector coordinate is a high-dimensional linear combination of upstream activations, which yields CLT-style mixing; and (iii) SGD with weight decay drifts toward maximum-entropy solutions on the resulting flat manifold, and a Gaussian is the maximum-entropy distribution at fixed mean and variance. Approximate Gaussianity of standard BARLOW TWINS marginals is therefore a *default state* of the optimisation, not a target of the loss. Figure 8(a), right column, shows distilled MATRYOSHKA sitting on this default state across all four prefix-multiplicity groups; the left column shows the naive method departing from it on the heavily-loaded early dimensions.

Two footprints from one mechanism. The m_j-loaded coordinates suffer in two coupled ways. **(i) Variance inflation.** Whether the m_j prefix gradients are independent (so that stationary variance scales linearly in m_j), aligned (quadratically), or partially correlated (intermediate) depends on inter-loss gradient correlation, which is loss-design-dependent and not predictable a priori. We therefore make only the safe ordering claim: *stationary variance σ_d^2 at coordinate j is monotone non-decreasing in m_j*. This predicts the strict ordering $\sigma_{[1,16]} > \sigma_{[17,32]} > \sigma_{[33,64]} > \sigma_{[65,128]}$ without committing to a specific exponent. **(ii) Marginal-shape distortion.** The m_j prefix losses place generically conflicting demands on the shared early coordinates. The conflict is *not* in representation space, where the constraints are nested and compatible: if the cross-correlation of $Z_{1:32}$ is exactly diagonal, its top-left 16×16 block is diagonal too, so whitening the wider prefix already whitens the narrower one. It arises instead from the *decoupled nonlinear projector heads*: each prefix loss whitens the output of its own head g_k ($g_{16} \neq g_{32}$, each mapping the shared prefix into the P-dimensional projector space), so whitening $g_{16}(Z_{1:16})$ and whitening $g_{32}(Z_{1:32})$ impose different constraints on the same early coordinates, sharpened by the $16 \to P$ rank bottleneck of the smallest head. Their stationary points generically conflict, and SGD oscillates on a noisy plateau where no single constraint is exactly satisfied. On this plateau, the implicit-Gaussian default no longer applies, and marginal distributions are free to take ad-hoc shapes (bimodal, heavy-tailed, skewed). The naive model's inflated σ_d step in Figure 8(c) and its non-Gaussian marginals on dims 0–15 in panel (a) are therefore not two phenomena: they are the same gradient-multiplicity pathology observed through two different statistics.

Observation 1 (Naive prefix-BT is unsupervised gauge-fixing by gradient counting; distilled MATRYOSHKA is supervised nested rate–distortion.)**.** Under prefix-BARLOW TWINS with prefix set $\mathcal{K}$ and multiplicity m_j defined by Equation (33), the optimisation is invariant under any orthogonal transformation that acts within each multiplicity-equivalence class $\{j : m_j = m\}$ but not across them. The remaining gauge freedom is therefore between equivalence classes, and is fixed at convergence not by any external semantic signal but by the optimisation dynamics: stationary variance σ_d^2 and marginal-shape distortion at coordinate j are both non-decreasing in m_j, and the implicit Gaussian regularity of single-prefix BARLOW TWINS is preserved on the $m_j = 1$ tail and progressively lost as m_j grows. In contrast, the MATRYOSHKA distillation objective $\sum_k \alpha_k\, d(h_k(s_{1:k}), t)$ ($\alpha_k = 1$ in all our experiments) supervises every prefix against a single fixed target $\boldsymbol{t}$, fixing the gauge through $\boldsymbol{t}$ and eliminating cross-prefix gradient conflict; it shares the multiplicity m_j, and hence a variance step, but inherits the implicit Gaussian regularity from the teacher.

Both predictions hold empirically. Figure 8 makes the test concrete. The σ_d ordering of Observation 1 is observed in panel c for both models: naive $\sigma_d \approx 1.85$ on dims 0–15 ($m_j = 4$) steps down to ≈ 1.5 at $d = 16$ ($m_j = 3$), ≈ 1.3 at $d = 32$ ($m_j = 2$), and ≈ 1.2 on the $m_j = 1$ tail, while the distilled model steps from ≈ 1.15 to ≈ 0.85, ≈ 0.80, and ≈ 0.75 at the same boundaries. The step locations match the prefix boundaries to within a single coordinate. The distilled step is the multiplicity footprint of the distillation loss itself and is a pure scale factor; the naive σ_d is 1.6–1.8$\times$ larger in every group and its tail does not return to the single-prefix scale. The marginal-shape distortion is observed in panel a and is what separates the two: naive dims 0–15 are heavy-tailed and bimodal, with the distortion weakening monotonically with m_j, whereas distilled marginals are Gaussian-like throughout all 128 coordinates.

Downstream consequences. Both footprints translate directly into the radar plots in Figure 8(b). On the 15 tasks of the ALPHAEARTH suite, the four configurations sit in the predicted order at

$d = 16$ and 32, with the two distilled variants becoming indistinguishable by $d = 64$: DISTILLED-FIRST > DISTILLED-RANDOM > NAIVE-FIRST > NAIVE-RANDOM. The FIRST-VS-RANDOM gap diagnoses whether a method has acquired a *real* coordinate ordering: it is small for distilled MATRYOSHKA (because random subsets of a faithfully compressed code remain informative) and large for naive MATRYOSHKA-BARLOW TWINS (because the apparent advantage of FIRST-D over RANDOM-D comes from the $m_j = 4$ over-loading of dims 0–15 rather than from genuinely ordered semantic content). The gap is most pronounced at $D = 16$, where the gradient-multiplicity asymmetry is extreme, and narrows at $D = 32, 64$ as more tail dimensions enter the prefix and the m_j gap closes.

Composite scores behind Figure 8(b). Table 21 reports, averaged over the seven balanced-accuracy classification tasks plotted in Figure 8(b) (Africa crop mask, GLanCE, LCMAP-LC, LCMAP-LU, Descals oil palm, LUCAS-LC, US trees), the mean prefix scores, chance-adjusted as in Section 3.1.

Table 21: Composite classification score per MATRYOSHKA prefix behind Figure 8(b), for distilled MATRYOSHKA and the naive MATRYOSHKA-BARLOW TWINS baseline, under FIRST-d and RANDOM-d prefixes.

	d=16		d=32		d=64	
Method	FIRST	RAND	FIRST	RAND	FIRST	RAND
Distilled MATRYOSHKA	0.486	0.472	0.529	0.511	0.543	0.545
Naive MATRYOSHKA-BARLOW TWINS	0.441	0.382	0.492	0.453	0.519	0.501

Distilled FIRST and RANDOM differ by at most 0.018 at every prefix dimension and become indistinguishable by $d = 64$. Naive FIRST exceeds naive RANDOM by 0.059 at $d = 16$, 0.039 at $d = 32$, and 0.018 at $d = 64$; the gap narrows as the $m_j = 4$ early-coordinate cluster contributes a smaller fraction of the prefix, exactly as Observation 1 predicts. The distilled-vs-naive gap at FIRST-d also contracts with dimension (0.045, 0.037, 0.024 for $d = 16, 32, 64$), but the FIRST-VS-RANDOM gap within the naive model is the more diagnostic quantity, since it isolates whether the apparent FIRST-d advantage reflects ordered semantic content or merely gradient multiplicity.

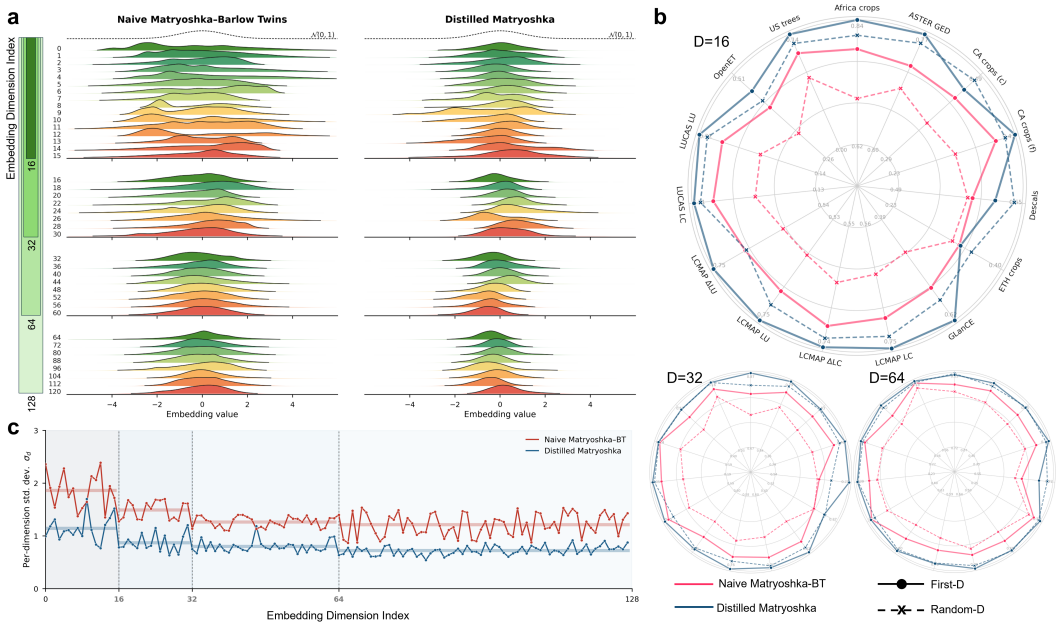

Figure 8: Why naive MATRYOSHKA-BARLOW TWINS fails. **(a)** Per-dimension marginals for the naive method (left) and distillation (right), grouped by prefix-loss multiplicity m_j; reference $\mathcal{N}(0, 1)$. **(c)** Per-dimension standard deviation σ_d; dashed lines mark the prefix boundaries $d = 16, 32, 64$. **(b)** Downstream radar at prefix sizes $D \in \{16, 32, 64\}$: naive vs. distilled, FIRST-D vs. RANDOM-D (five seeds).

Distillation supplies the missing signal on two fronts. Distillation against a fixed teacher target repairs both footprints separated by Observation 1. **(i) An external semantic ordering.** Each prefix $\boldsymbol{s}_{1:k}$ is supervised to reconstruct the full teacher embedding $\boldsymbol{t}$, not the teacher's own first k coordinates; the prefix becomes a k-dimensional rate–distortion bottleneck on a fixed target distribution, and later dimensions encode the residual. The gauge that prefix-BT fixed by gradient counting is now fixed by the teacher. **(ii) Inherited marginal regularity.** The $2\,B$ teacher was trained with single-prefix BARLOW TWINS and therefore inherits its implicit Gaussian marginals. Under MSE-style reconstruction, the optimal compression of an approximately Gaussian target produces approximately Gaussian projections. Distilled prefix marginals look Gaussian because they inherit the teacher's distribution, not because we asked them to. The same pattern applies whenever a self-supervised objective leaves a coordinate gauge and a nested-representation requirement is imposed afterwards: the gauge must be fixed by an external target distribution. *Self-supervision learns a representation subspace; distillation learns an ordered code.*

V.1 DISTILLATION TURNS PREFIX LEARNING INTO ORDERED COMPRESSION

A direct consequence is that the student's first 16 dimensions need not match, and typically should not match, the teacher's first 16 coordinates. The teacher's coordinates are arbitrary; the student's first 16 are a *learned compressed code* for the teacher's full d_T-dimensional embedding. We confirm this empirically: distilled MATRYOSHKA prefixes $\boldsymbol{s}_{1:16}, \boldsymbol{s}_{1:32}, \boldsymbol{s}_{1:64}$ outperform the corresponding teacher prefixes $\boldsymbol{t}_{1:16}, \boldsymbol{t}_{1:32}, \boldsymbol{t}_{1:64}$, despite the student being far smaller (Appendix W.3).

W ABLATIONS

W.1 DISTILLATION VERSUS SAME-SIZE FROM-SCRATCH TRAINING

The production recipe trains a large teacher and distils it, which raises the obvious control: is the teacher needed at all, or could a $44\,M$ encoder, the size of the deployed Tessera v2-2B-L student, be trained from scratch to the same quality? We test this directly. Taking the identical $44\,M$ architecture and the BARLOW TWINS objective of Section 4, and giving it the teacher's own data budget, we train it from scratch to convergence. It reaches an ALPHAEARTH suite composite of 0.527, below even Tessera v1 (0.541) and well below the 0.593 of the distilled student of the same size (Table 22). This is the comparison the per-encoder-best curve of Figure 2f already traces out: that curve is the from-scratch score of pixel-wise encoders across the sweep, and a $44\,M$ encoder sits on it well below both the distilled student and the teacher. The 0.066 gap is what distillation buys. The 2B teacher learns a representation that a same-size encoder cannot recover on its own, and distillation transfers most of it into $44\,M$ parameters, whose own from-scratch ceiling is 0.527 (the teacher itself scores 0.608). Train-large-then-distil is therefore not a convenience but a requirement: $44\,M$ parameters are enough to *host* the teacher's representation but not to *discover* it.

Table 22: **A same-size encoder cannot be trained from scratch to the distilled student's quality.** ALPHAEARTH suite composite for a $44\,M$ encoder trained from scratch with BARLOW TWINS on the teacher's data budget, versus distilled from the 2B teacher; the 2B teacher is an upper reference.

$44\,M$ encoder	ALPHAEARTH suite composite
From scratch (BARLOW TWINS, teacher data budget)	0.527
Distilled from the 2B teacher (Tessera v2-2B-L)	**0.593**
2B teacher, upper reference	0.608

W.2 COORDINATE-EXCHANGEABILITY TEST

Appendix V predicted that naive MATRYOSHKA-BARLOW TWINS should exhibit a gradient-imbalance fingerprint at exactly the prefix boundaries $d = 16, 32, 64$, and Figure 8 confirmed that prediction both at the level of representation geometry and at the level of downstream task performance. The comparison against Tessera v1 below isolates *which symmetry is broken, and how*.

For each of three methods we evaluate $k = 16$ prefixes constructed three ways: FIRST-16, RANDOM-16, LAST-16.

- *Tessera v1* (Feng et al., 2026) produces nearly identical performance from all three prefix variants, consistent with the rotation symmetry of BARLOW TWINS: there is no preferred coordinate ordering, and any 16-dimensional subspace is essentially equivalent.
- *Naive* MATRYOSHKA-BARLOW TWINS produces the ordering FIRST-16 > RANDOM-16 > LAST-16, i.e., the prefix losses do break the rotation symmetry, but the absolute quality of FIRST-16 is not better than Tessera v1's, and is sometimes worse. This is the gradient-imbalance signature visualised in Figure 8: dimensions 0–15 are overconstrained rather than informationally privileged.
- *Distilled* MATRYOSHKA produces a clean ordering with strong absolute quality at small k: its FIRST-16 prefix is the best of any method, ahead of the corresponding Tessera v1 prefix and markedly ahead of the teacher's own coordinate prefix, and at $k=128$ it converges to Tessera v1 and the teacher coordinate prefix.

Naive MATRYOSHKA-BARLOW TWINS does break the rotation symmetry of BARLOW TWINS, but it breaks it in the wrong way; only distillation supplies the external target distribution that lets prefix learning function as ordered compression rather than as gradient-imbalance overtraining.

Protocol. We evaluate three prefix-construction protocols on each model's 128-dimensional output: FIRST-16 (the first 16 coordinates), RANDOM-16 (a uniformly sampled 16-coordinate subset, averaged over five seeds), and LAST-16 (the final 16 coordinates). Performance is the composite chance-adjusted downstream score (Section 3.1) over the seven classification datasets of Figure 8(b); the $k=128$ column reports the same composite at full dimensionality (Table 23).

Table 23: Coordinate-exchangeability test (composite chance-adjusted downstream score over the 7 classification datasets of Figure 8(b), ↑). Best FIRST-16 and best $k=128$ in bold.

Method	FIRST-16	RAND-16	LAST-16	$k=128$
Tessera v1	0.470	0.473	0.465	0.559
Naive MATRYOSHKA-BARLOW TWINS	0.441	0.382	0.360	0.537
Teacher coordinate prefixes	0.430	0.427	0.420	**0.565**
Distilled MATRYOSHKA (Ours)	**0.486**	0.472	0.458	0.553

In Table 23, naive MATRYOSHKA-BARLOW TWINS separates FIRST-16 from RANDOM-16 by 0.059, confirming that prefix losses break coordinate exchangeability, but its FIRST-16 score (0.441) remains *below* the distilled RANDOM-16 score (0.472), showing that the naive FIRST advantage is not evidence of useful ordered compression. Tessera v1 is approximately coordinate-exchangeable, with FIRST/RANDOM/LAST all near 0.47. Teacher coordinate prefixes are also weak at $k = 16$ (0.430) because the teacher's coordinate basis is not itself ordered by downstream utility. Distilled MATRYOSHKA obtains the strongest FIRST-16 score (0.486) and a near-best full-dimensional score (0.553), because its ordering signal comes from reconstructing a fixed teacher target rather than from repeated self-supervised losses on early coordinates.

W.3 TEACHER PREFIX VERSUS STUDENT PREFIX

A direct prediction of Appendix V.1 is that the *student*'s first k dimensions should be a learned compression of the teacher's full embedding, not an imitation of the teacher's first k coordinates. The learned student prefix $\boldsymbol{s}_{1:k}$ is supervised to reconstruct the full teacher embedding $\boldsymbol{t}$, whereas a teacher coordinate-prefix $\boldsymbol{t}_{1:k}$ is merely the first k coordinates of $\boldsymbol{t}$ and has no reason to be an optimal k-dimensional code: the teacher's coordinate basis is not ordered by downstream utility.

We fit reconstructors from $\boldsymbol{s}_{1:k} \to \boldsymbol{t}$ and from $\boldsymbol{t}_{1:k} \to \boldsymbol{t}$ and report cosine similarity, MSE, and linear R^2 to the full teacher target (Table 24). At every k the student prefix is a markedly better code for $\boldsymbol{t}$ than the teacher coordinate-prefix; the gap is largest at $k = 16$ (cosine 0.842 vs. 0.684, R^2 0.621 vs. 0.358) and contracts with k (the teacher is d_T-dimensional, so even at $k=128$ the teacher coordinate-prefix remains a strict prefix). Spectral diagnostics (normalised effective rank, CKA, and Procrustes alignment to the full teacher) tell the same story and are reported in Appendix R.

Table 24: Prefix-to-teacher reconstruction: fit quality of the student prefix $\boldsymbol{s}_{1:k}$ vs. the teacher coordinate-prefix $\boldsymbol{t}_{1:k}$ as codes for the full teacher embedding $\boldsymbol{t}$.

k	Source	Cosine to $\boldsymbol{t}$ ↑	MSE ↓	Linear R^2 ↑
16	student $\boldsymbol{s}_{1:16}$	0.842	0.178	0.621
	teacher $\boldsymbol{t}_{1:16}$	0.684	0.304	0.358
32	student $\boldsymbol{s}_{1:32}$	0.905	0.118	0.759
	teacher $\boldsymbol{t}_{1:32}$	0.764	0.238	0.512
64	student $\boldsymbol{s}_{1:64}$	0.956	0.062	0.884
	teacher $\boldsymbol{t}_{1:64}$	0.863	0.146	0.692

W.4 Teacher size

We ablate teacher size by repeating the distillation pipeline from teachers of the same architectural family at different widths and measuring student performance. As predicted by the scaling laws, larger teachers yield better students at fixed student size, and they also yield better prefix retention: a larger teacher provides a richer target distribution against which early-coordinate compression has more to exploit (Table 25). Both the full-dimensional composite and the d=16 retention ratio rise with teacher size, from 91.1% retention for the 500 M teacher to 92.1% for the 2 B teacher used in all reported results. This is the load-bearing reason "train large" precedes "distil flexibly": the teacher absorbs the pretraining compute, while the student learns a deployable ordered code over the teacher's representation distribution, and a weak teacher cannot be papered over by clever distillation.

Table 25: Teacher size and prefix supervision: Tessera v2-L student composite at d=16 and d=128, and the d=16 retention ratio, as a function of teacher size.

Teacher size	d=16	d=128	d=16 retained
500 M	0.513	0.563	91.1%
1 B	0.534	0.584	91.4%
2 B (reported)	0.546	0.593	92.1%

W.5 Storage and deployment trade-offs

For embedding-as-data, the storage scaling is multiplicative: $d = 16$ uses $1/8$ of the storage of $d = 128$, and combined with int8 quantisation (Feng et al., 2026) the saving relative to fp32 128-D embeddings is 32×. At planetary scale, this changes whether global annual embeddings fit on a workstation or require a cloud-hosted store. Table 26 reports the per-prefix composite for the full student family, Table 27 the retention ratios for Tessera v2-2B-L, and Table 28 the corresponding storage footprints: each prefix corresponds to a global annual storage budget at 10 m land scale, and the Matryoshka layout turns d into a deployment knob that users set against their I/O and storage budget without retraining, preserving the same model family, coordinate convention, and downstream interface. d=16 retains ∼92% of d=128 performance on average with negligible degradation on classification-like tasks, while d=64 recovers nearly the full composite at half the storage and is the recommended default when regression or fine-grained class distinctions are central.

Table 26: Composite downstream score over the 15 AlphaEarth suite tasks for each distilled student and Matryoshka prefix.

Student	d=16	d=32	d=64	d=128
Tessera v2-2B-N	0.516	0.542	0.555	0.560
Tessera v2-2B-S	0.531	0.559	0.572	0.577
Tessera v2-2B-M	0.540	0.567	0.581	0.586
Tessera v2-2B-L	0.546	0.574	0.588	0.593

Table 27: Composite downstream score per MATRYOSHKA prefix for Tessera v2-2B-L, mean over the 15 ALPHAEARTH suite tasks.

Prefix d	16	32	64	128
Composite	0.546	0.574	0.588	0.593
Retained vs d=128	92.1%	96.8%	99.2%	100%

Table 28: Storage–accuracy trade-off across MATRYOSHKA prefixes for Tessera v2-2B-L (int8). Global annual storage assumes $\approx 1.5 \times 10^{12}$ land d-pixels at 10 m (Appendix B.8).

Prefix d	Storage/d-pixel	Global annual storage	Composite	Recommended use
16	16 B	∼22 TiB	0.546	Edge / on-device; cheapest broad-coverage product
32	32 B	∼44 TiB	0.574	Regional, storage-constrained pipelines
64	64 B	∼87 TiB	0.588	Balanced product; near-full at half storage
128	128 B	∼175 TiB	0.593	Research / maximum-accuracy workflows

X EMBEDDING ARTEFACTS AND TEMPORAL STABILITY

This appendix expands the embedding-quality summary of Section 5.1.

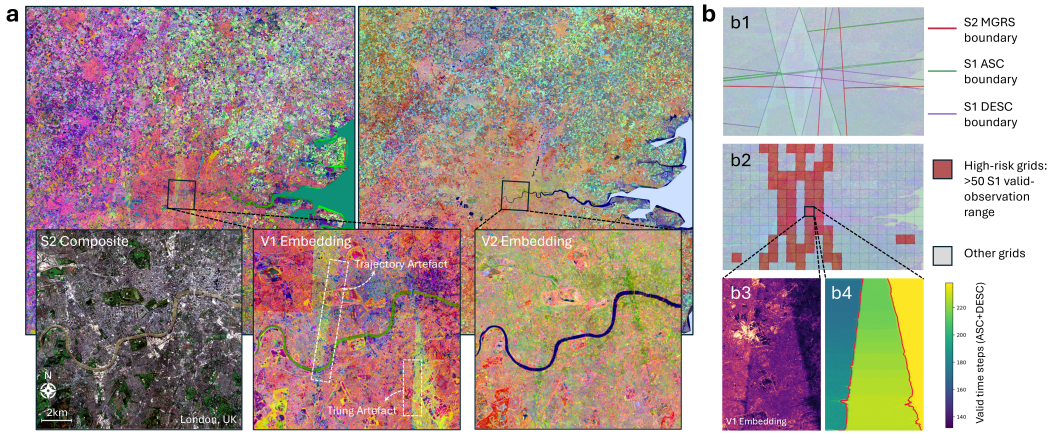

Figure 9: Acquisition-driven artefacts over London. **(a)** RGB renderings of v1 and v2 embeddings; v1 shows along-track striping and seam discontinuities, v2 does not. **(b)** Diagnosis: (b1) Sentinel-2 MGRS and Sentinel-1 ascending/descending boundaries overlaid on the embedding field; (b2) high-risk cells on the 0.1° inference grid; (b3) a sharp v1 discontinuity in one such cell, coincident with (b4) a strong gradient in Sentinel-1 valid-observation count.

Removal of acquisition-driven artefacts. Tessera v1's embeddings, rendered as RGB false-colour maps, exhibit visible *trajectory* artefacts (along-track striping) and *tiling* artefacts (seam discontinuities at processing boundaries): the embedding inherits, rather than absorbs, sensor-acquisition nuisance (Figure 9a, insets). The v2 student embeddings do not show these artefacts on the same regions, while preserving genuine geographic structure: the River Thames, the urban core, and the surrounding countryside all remain crisp. We attribute the improvement to two changes: the larger teacher learns a more sampling-invariant temporal representation under BARLOW TWINS (aided by multi-scale training and full-observation inference), and distillation transfers this invariance to the deployed students rather than forcing them to relearn it.

The artefacts are acquisition-driven and predictable. The artefacts in v1 are not random noise; they are systematic and tied to sensor-acquisition geometry, which we establish through a four-step diagnosis (Figure 9b). *First*, overlaying Sentinel-2 MGRS boundaries and Sentinel-1 ascending/descending acquisition boundaries on the embedding field shows that many visible discontinuities align with this geometry rather than with land-cover boundaries (Figure 9(b1)). *Second*, the 0.1° grid we use for global precomputation lets us flag high-risk cells in advance: those that intersect a Sentinel-

1 tangent boundary *and* whose within-cell range of valid Sentinel-1 observation counts exceeds 50 (Figure 9(b2)). *Third*, a representative high-risk cell indeed contains a sharp local discontinuity in the v1 embedding (Figure 9(b3)). *Fourth*, that same cell exhibits a strong spatial gradient in Sentinel-1 valid-observation count, and the embedding discontinuity is spatially coincident with it (Figure 9(b4)). The mechanism is that regions receiving different numbers of valid Sentinel-1 observations induce different embedding statistics in v1, which surface as acquisition-aligned seams. Tessera v2's full-observation bucket inference (Section 4) removes the dependence on a fixed observation budget, which is why the same regions are artefact-free under v2.

Inter-annual stability across 2017–2025**.** Annual embeddings are intended to support time-series analyses (disturbance detection, land-cover change). A good annual representation should be stable on stable land cover and respond cleanly to actual change, returning to a clean baseline once a transient event has passed. On stable land-cover regions (intact forest, urban cores) the consecutive-year cosine distance between v2 embeddings is markedly lower than v1's; on disturbance pixels (logging area) both models register the event, but only v2 returns to a clean post-disturbance baseline, while v1 remains noisy (Figure 1d).

Y BROADER IMPACT

Lower-cost EO representations expand who can build planetary-scale analyses (ecologists, smallholder agriculture programmes, climate adaptation efforts, disaster-response teams) without proprietary infrastructure. The same representations can in principle support surveillance and sensitive land-use monitoring. Our release policy is open under a CC0 license: we release the distilled pixel-wise students and the released data product, as well as the $2B$ teacher checkpoints for full reproducibility and extensibility.