Operational HPC lessons learnt training the Tessera geospatial foundation model series

Zhengpeng Feng, Mark T. Elvers, Sadiq Jaffer, Michael W. Dales, Malcolm Scott, Pedro Sousa, Will Tebbutt, Richard E. Turner, David A. Coomes, Srinivasan Keshav and Anil Madhavapeddy*
{zf281,mte24,sj514,mwd24,pmms2,dac18,sk818,mas90,avsm2}@cam.ac.uk
University of Cambridge
Cambridge, United Kingdom

1 Overview

Tessera is a pixel-wise Earth-observation foundation model that fuses irregular satellite optical and radar time series into a single annual embedding[1] for every 10 m land pixel on the planet. Researchers can use these open, analysis-ready vectors for classification, segmentation, and regression tasks without repeating satellite preprocessing or model inference. Applications span ecology, agriculture, forestry, biodiversity, biomass estimation, and weather [21, 22, 43, 56, 57]. Within a year of release, Tessera has attracted adoption and embedding requests from researchers and organisations worldwide [32].

At its core, Tessera takes a large number of isolated time-series observations and reduces them to compact annual embeddings. This separation between learning a shared representation once with large-scale self-supervised training, and then applying it globally to generate easy-to-use map tiles is what makes both the scientific payoff worthwhile and the infrastructure challenge significant.

It is unusual to train a state-of-the-art geospatial foundation model of this scale (§1.2) from scratch in an academic setting. We are a University of Cambridge research group without dedicated access to a cluster at the scale required for planetary computation. We therefore assembled a succession of national, institutional, developer-cloud, and commercial compute allocations to train and publish Tessera, including large runs using Intel (Dawn), NVIDIA (Isambard) and AMD (Vultr) GPUs. This report is aimed at research-computing practitioners who also need to move data-intensive models between heterogeneous systems. We first describe the scientific payoff and workload, then report what worked and what failed on each platform, before distilling cross-provider recommendations, energy-accounting lessons, and future requirements (§2).

1.1 Scientific goals

Our scientific objective is to turn irregular, cloud-affected satellite observations into a consistent description of land-surface state. Unlike an individual image, an annual Tessera embedding captures seasonal optical and radar behaviour at each location. This representation allows one model output to support many environmental questions while reducing the labelled observations and task-specific computation required by each study.

Tessera v1 (Dec 2025) established reusable surface representations by generating label-efficient, 128-dimensional embeddings from Sentinel-1 and Sentinel-2 time series. It closely matched or exceeded task-specific and foundation-model baselines across classification and segmentation tasks [21, 29].

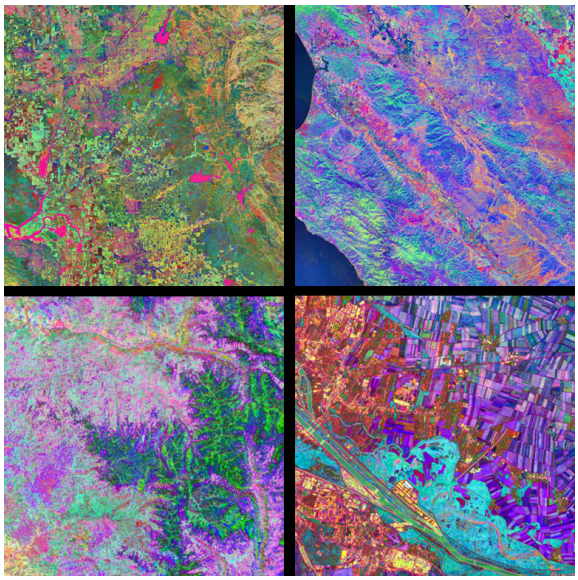


Figure 1: Examples of false-colour maps of Tessera v1 [19].

Tessera v1.1 (June 2026) added year-on-year temporal stability. The retrained model maintained temporal stability across years, reduced artefacts in areas with few observations, and extended coastal coverage to support more marine tasks [34].

Tessera v2 (July 2026) scaled up quality and deployability significantly. Across a suite of 29 environmental tasks, a 21-million-parameter v2 model achieved the strongest aggregate result among all products evaluated. It supports a Matryoshka representation that retains high performance with a subset of dimensions [22].

Tessera supports joint environmental tasks beyond satellite mapping. A wide array of downstream applications address topics such as biomass, crops, fungal biodiversity, and tree species identification [5, 20, 28, 30, 55–57]. In probabilistic weather downscaling, a Tessera embedding supplied sub-grid surface information to the Aardvark weather-prediction system, improving temperature and wind predictions at stations withheld in both space and time [4, 43].

All principal outputs are openly accessible. We released weights [25], source code and a versioned, actively maintained GeoTessera client [24, 51], and planet-wide embeddings through public object storage [32]. Users worldwide can therefore leverage the pretraining without repeating its computational cost, as demonstrated by downstream environmental studies [5, 23, 28, 43].

*Corresponding author: Anil Madhavapeddy, avsm2@cam.ac.uk, 22nd Sep 2026
[1] An "embedding" is a compact numerical description of surface state.

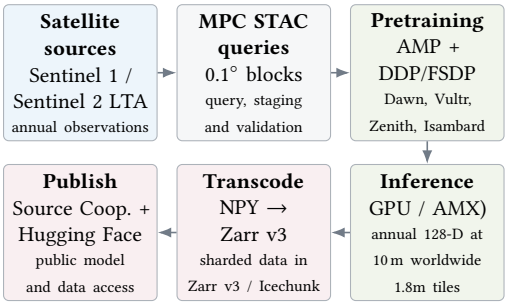


Figure 2: The Tessera computation path [13, 21, 32].

1.2 Computational workload

Figure 2 summarises our processing pipeline [13, 15, 22]. The separation between pretraining, planetary inference, and public hosting at scale drives our operational constraints (§2). The training stage learns the shared representation, and inference applies the frozen model to new locations around the world. Our "embeddings as data" approach publishes those inferred outputs as a reusable geospatial layer so users can integrate directly into existing workflows.

The optical input is Sentinel-2 Level-2A (L2A) surface reflectance, while the radar input is Sentinel-1 radiometrically terrain-corrected (RTC) imagery. Both datasets come from the Microsoft Planetary Computer (MPC) [38, 42]. We locate imagery through its SpatioTemporal Asset Catalog (STAC) interface, an open standard for describing and querying spatiotemporal geospatial assets [40].

Training then uses automatic mixed precision (AMP) with PyTorch Distributed Data Parallel (DDP) or Fully Sharded Data Parallel (FSDP) [58]. Inference of individual embedding tiles currently runs on GPUs; a CPU path using Intel Advanced Matrix Extensions (AMX) is planned. The resulting vectors are quantized to 8-bit integers and published in both NPY format and Zarr, a chunked array format that supports streaming HTTP input/output [33].

1.2.1 The pretraining challenge. For every 10 m pixel, the input pipeline queries a full year of imagery for that region. It downloads Sentinel-2 scene classification masks and ten optical bands for valid dates, collects both Sentinel-1 viewing directions, and randomly selects 40 observations from each sensor [13]. The discovery, reprojection, masking, and batch assembly code stages need to keep each GPU supplied to avoid stalls that would reduce hardware utilisation.

V1 was trained on 800 million annual pixel records with a global shuffle of 2 TB of prepared data to prevent batches of similar neighbouring pixels from producing unstable training. One pass through this corpus at a batch of 32k samples per optimisation step over 16 GPUs consumed approximately 6,200 MI300X GPU-hours. While the deployable v1 encoder contained only 45.7 million parameters, the training-only projector was much larger at 1.39 billion parameters. This means that the deployed model size understated training memory and communication demand [21].

V2 increased the training corpus to approximately 4.2 billion pixel-years spread over 512 GPUs [22]. Once data images were staged, measurements showed that training time was predominantly spent on the GPU, a process that took significant optimisation work but paid off with efficiency.

1.2.2 The global inference challenge. Once the pretraining is complete, we need to generate Tessera map tiles for the world. Global inference reads and validates far more unique remote imagery than the earlier training runs, since we have to cover the whole planet. Unfortunately, rate-limited requests from our satellite providers can resemble missing objects, and so we had to carefully triage errors so that an omitted observation did not create visible artefacts.

We estimated that we need approximately 2 H100 GPU-years for a single year's pass with our v2 model [22]. Each pixel initially produces a 128-dimensional, 32-bit floating-point vector occupying 512 bytes, which we quantise to 8-bit integers with one 32-bit scale per pixel, reducing each embedding to 132 bytes. The 2017–2025 target therefore contains approximately 1.3×10^{13} 10 m land pixel-years and an unreplicated quantised payload of roughly 1.7 PB, before metadata and storage overhead [21, 33]. This exceeded available storage in our University and departmental storage arrays by an order of magnitude.

1.2.3 The long-term hosting challenge. Once we have done the inference locally, we need to publish it so that others can download it. The models themselves are relatively small, and we publish their weights on the Hugging Face Hub [25]. However, the hosting challenge centres around the map tiles themselves which take up to 1.7 PB per model. We also had to account for multiple models being developed concurrently while users migrated from older ones at different paces. Therefore, the logical Zarr dataset before replication and service metadata is around 3–4 PB.

Even this smaller volume is impractical to serve from University infrastructure, so we need well-connected and affordable hosting that is accessible worldwide. These embeddings are also inputs to environmental studies that may run for decades. Hosting must therefore eventually lead to a live archive as versioned products that remain discoverable and usable as providers, software, and research groups change, rather than being dependent on one organisation's storage account. Collaborative geospatial archives provide a useful precedent [18, 36] but at the petabyte scale.

1.3 Infrastructure requirements

The earlier scale estimates (§1.2) translate into the following infrastructure requirements. The 2017–2025 product is petabyte-scale, with several model versions under active development [21, 32].

(R1) **Sustained data access.** Training and inference require concurrent access to petabyte-scale public Sentinel archives through Microsoft Planetary Computer without a shared egress identity becoming the throughput limit.

(R2) **Distributed training.** Scaling requires GPU support for mixed-precision, high-bandwidth collectives, recoverable checkpoints, and predictable access to hundreds of accelerators.

(R3) **Planetary inference.** Parallel inference requires CPU for discovery and preprocessing, GPUs for current inference (with an AMX CPU path planned), and a shared data path that can keep compute flowing while performing input validation.

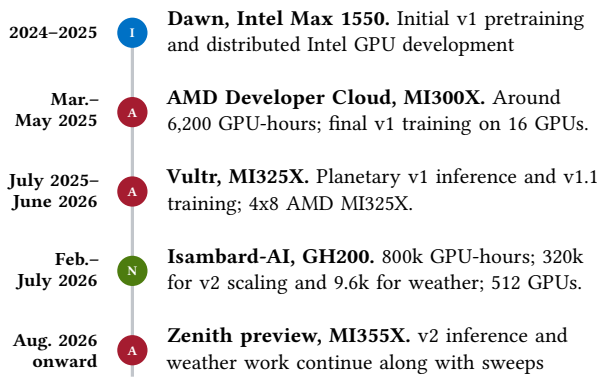
(R4) **Storage and publication.** Training requires at least 40 TB of fast working space, while publishing the multi-year embedding product requires petabyte-class durable capacity. Both tiers need high inode capacity; public object storage also has to remain separate from the research working tier.

(R5) **Observability and accounting.** Development requires interactive access, CPU/GPU performance counters, and job-level records for compute, energy, failure time, and network metrics.

(R6) **Agent-compatible operation.** Increasing use of coding agents requires machine-readable documentation, CLI tools, scoped credentials, structured telemetry, and auditable access to schedulers.

2 Compute, networking and storage experiences

The sequence of Intel (§2.1), AMD (§2.2, §2.4), and NVIDIA (§2.3) deployments is shown below. The transitions followed scientific priorities, expiring allocations, and operational constraints rather than a predetermined plan. We also describe the Cambridge Ceph storage (§2.5) and conclude with overall recommendations.



2.1 Dawn for initial v1 pretraining

Dawn is the University of Cambridge's 256-node Intel GPU system and our first distributed accelerator platform [19, 52].

2.1.1 What worked well. Dawn provided an essential early-access system for the v1 training design via an Intel distributed-GPU path. Each node combined two 48-core Xeon Platinum 8468 CPUs, 1 TiB of host memory, four 128 GB Intel Data Center GPU Max 1550 packages connected by the Xe-Link device fabric, and quad HDR200 InfiniBand networking. We built PyTorch with the Intel Extension for PyTorch (IPEX), and brought up the oneAPI Collective Communications Library (oneCCL) across nodes. This made the training code portable and the later AMD port rapid [19, 52].

2.1.2 What constrained us. The default configuration exposed the two tiles in each Max 1550 package as separate logical devices, giving eight 64 GB devices per node. Merging tiles did recover a single address space, but introduced large non-uniform memory-access (NUMA) penalties. Treating them separately required the Slurm batch scheduler to account for eight devices rather than four packages. This in turn meant that Message Passing Interface (MPI) launches could fail intermittently. The default oneCCL IPC path was unreliable, and so careful manual tuning was needed for FSDP.

We disabled IPEX optimisations because they destabilised distributed training after the driver allocated several terabytes of virtual address space and crashed the node. Single-node training took about two months to bring up, followed by a month for multi-node training. Restricted performance counters prevented us from separating HBM stalls, kernel occupancy, and other overhead [19].

Recommendation 1

If ML workloads are anticipated, ship a validated PyTorch+MPI with examples for the site's tile topology. Provide production-equivalent interactive nodes and job-scoped performance counters. *Addresses* (R2) (R5)

2.2 AMD and Vultr for v1 training and inference

The AMD Developer Cloud provided a three-month MI300X allocation from March 2025, bridging Dawn's Intel environment to ROCm. After a one-month gap, hosted Vultr MI325X instances provided continuity for v1 inference and v1.1 training [2].

2.2.1 What worked well. Our PyTorch pipeline was brought up on AMD's ROCm GPU software stack within 2 hours of gaining access to the host, and ran across eight MI300X GPUs by the following morning. There was sufficient data throughput to keep the GPUs supplied, so most elapsed training time was GPU computation.

In two hyperparameter-search runs with identical data, model size, and parameters, MI300X reduced elapsed time relative to Dawn from 127 to 51 minutes, while final loss and validation F1 remained effectively unchanged. Both runs used AMP. The matched measurements are shown below; Dawn's percentages are relative to one 64 GB logical tile and MI300X's to one 192 GB GPU.

Observed metric	Dawn	MI300X
Allocated HBM	~46 GB (72%)	~27 GB (14%)
Elapsed time	127 min	51 min
Time per step	927 ms	325 ms
Final loss	29.59	29.65
Best validation F1	0.7484	0.7498

PyTorch graph compilation through `torch.compile` improved MI300X throughput by 1.46× in default mode and 1.60× with maximum autotuning [2, 19].

2.2.2 What constrained us. This was an application comparison rather than a peak-hardware test: Dawn used split tiles, whereas MI300X provided 192 GB HBM3 and 5.3 TB/s package bandwidth, together with graph compilation [1, 19]. The only migration blocker was a `torch.compile` `assert_size_stride` dimension assertion; an upstream workaround resolved it [41].

The transition to Vultr was otherwise smooth, but planetary inference transferred so many Sentinel inputs from Microsoft Planetary Computer (MPC) that source bandwidth and rate limits became the dominant constraint, leading to GPU underutilisation. In addition, one GPU in an eight-GPU host would occasionally disappear, consistent with a hardware or thermal fault; the inference stack therefore had to detect the reduced device set and adjust. Tile-level logs and explicit validation prevented throttled or missing inputs from becoming embedding artefacts [16].

Recommendation 2

Publish versioned containers with tested PyTorch and ROCm combinations so that workloads can move between providers without a full software stack rebuild. *Addresses* (R2) (R3)

2.3 Isambard-AI for v2 scaling and weather

Our Isambard-AI [37] access followed a successful proposal to UKRI's AIRR "AI for Science" compute call, which offered 200,000–1,000,000 GPU-hours for a six-month project [48]. The award arrived with little lead time, so we brought the training stack into production within days. It enabled the large-scale v2 scaling study and weather applications (§1.2). We also established a working collaboration with NVIDIA to help us optimise the kernels before deploying the multi-month training runs.

2.3.1 What worked well. The 800,000 GPU-hour allocation made all three goals feasible. First, tightly coupled runs on up to 512 GH200s established the v2 scaling curves and supported the large-teacher training. Second, the resulting measurements informed the final v2 configuration and its quality improvements. Third, the weather study used many independent one-GH200 runs, repeated over three random seeds, and consumed about 9,600 GPU-hours. Every compute node had a public Internet Protocol (IP) address, so parallel workers could retrieve remote satellite data without collapsing onto one facility-wide source identity [6, 22, 37, 43].

2.3.2 What constrained us. The Isambard policy limited jobs to 24 hours and the limit could not be raised despite a support request. Checkpointing made interruptions recoverable, but a resumed large job could wait hours to days for another allocation. One of our runs requiring about 40 days of on-GPU execution consequently took substantially longer in wall-clock time.

The software-package availability for nodes' ARM64 architecture led to some hangs in the NVIDIA Collective Communications Library (NCCL) and CUDA process-fork deadlocks. This added some engineering overhead, but we attribute this to one-off growing pains as the ARM libraries are rapidly maturing. V2 also required at least 40 TB of working storage, and an early per-pixel layout exhausted our inode budget. We adopted a sharding strategy to reduce our raw file usage.

Recommendation 3

Retain routable node networking and add reserved multi-day reservation windows or priority-resumed job chains for large checkpointed runs. *Addresses* R2 R5

2.4 Zenith for v2 inference and change models

Zenith is Cambridge's preview-stage AMD system, where we tested v2 inference and follow-on training on MI355X accelerators with shared Lustre storage [53]. The measurements below are operational indicators rather than a like-for-like benchmark against MI300X: accelerator throughput was high, but external input delivery constrained end-to-end progress. An experimental change-detection model also exhausted its inode quota; a support request led to a rapid increase.

2.4.1 What worked well. Preview status made capacity substantially easier to obtain, and MI355X compute and the internal storage path performed well. A CPU-only tier downloaded and preprocessed imagery into 3.1 TB staged on the shared Lustre parallel filesystem, after which GPU workers ran at capacity. The distillation workload measured between 1,240–1,610 trillion floating-point operations per second using the bfloat16 format for 4096^3 and 8192^3 matrix products. The ROCm Collective Communications Library (RCCL) reached 364 GiB/s for an intra-node all-reduce (the operation that combines distributed gradients) and 113 GiB/s across two nodes. Inference completed in about 45 seconds per tile per GPU, and 48 workers delivered 244–258 tiles per hour [17].

2.4.2 What constrained the data path. All CPU and GPU nodes shared external network address translation (NAT), so MPC saw one source address and rate limited us. A tile took approximately 15 minutes to download from Zenith but less than 45 seconds to infer. Staging with 64 four-core jobs across 32 CPU nodes delivered 208 tiles/hour. Approximately 113 such nodes would have been needed to feed one eight-GPU node, but only 37 existed [17].

2.4.3 What constrained multi-node execution. A whole-node minimum allocation and an 8-minute configuration phase reduced efficiency for small tests, consistent with nodes being powered on and off between allocations. Our early inference used MI300-targeted kernels; a native ROCm 7 wheel for the MI355X `gfx950` architecture target existed but was difficult to discover. Multi-node training initially failed when nodes exposed different active InfiniBand ports, but disabling RCCL's preset topology matching forced successful generic graph discovery. A substantial fraction of preview nodes being unavailable also delayed four-node placement.

Recommendation 4

Give ingestion workers public IPv4 or routed IPv6, or provide independent egress pools. Validate homogeneous host-channel-adapter configuration and document RCCL fallbacks for heterogeneous topologies. *Addresses* R1 R3

2.5 Cambridge Ceph and public hosting

Tessera's source data, intermediate products, and embeddings exceeded the storage available to the project. We therefore needed a local staging tier for source data, intermediate products, and embeddings within the Cambridge Computer Laboratory. The group received donated machines and purchased SSDs for them shortly before supply shortages drove storage costs up by roughly 4–6 times [45, 46].

We chose Ceph because its distributed object and filesystem design and CRUSH placement algorithm support heterogeneous, incrementally expanded clusters [54].

What worked well. The team built an in-house Ceph distributed-storage cluster from reused Dell R640 servers and a heterogeneous stock of solid-state drives (SSDs). The first migration stage held 209 TB on seven nodes. The Cambridge cluster subsequently grew to 86 Object Storage Daemons (OSDs), each managing one storage device, across 20 hosts. This setup provided 600 TB raw and approximately 480 TB usable under 8+2 erasure coding, in which two parity shards protect every eight data shards. CephFS, Ceph's filesystem interface, supplied separate model subvolumes with quotas and snapshots [11, 12].

What constrained us. Operating storage became a substantial research responsibility. The mixed machines and devices required manual OSD placement and balancing, as Ceph groups objects into

placement groups before its CRUSH placement algorithm maps them to devices. Our initially undersized placement-group count resulted in repeated data remapping. In later expansions we paused backfill so CRUSH could rebalance once and save SSD wear [9, 12].

A mid-2026 expansion obtained twenty 8 TB enterprise SSDs described as "the last available stock for some time". Rebuilding equivalent flash capacity in 2026 would not have been affordable within the same project budget [45, 46]. One reason local cluster storage is necessary is because most HPC access and cloud storage is typically time limited, so we had to regularly backup 10s of terabytes of primary data for downstream tasks (e.g. weather station readings for Aardvark [4], intermediate model weights, or map outputs). Migrating these between cloud and HPC clusters took significant time and bandwidth in between allocations.

Recommendation 5

Couple GPU awards to durable working storage and a supported route into public object storage. Research groups should not need to assemble and operate heterogeneous hardware to publish an open output. *Addresses* R4

Transition to public infrastructure. This could have been scaled in-house with additional TrueNAS/JBOD or Ceph capacity, but the Cambridge Computer Lab's capital, power, cooling and floor-space envelope made a public object store more practical. GeoTessera 0.10 [24, 32] moved these stores to Source Cooperative.

We first packaged the transcoding worker as a Docker container for local deployment and testing [35], then ran the same image in Amazon Fargate Spot in the source-data (`us-west-2`) region. Moving so much data from Cambridge to AWS came with a significant cost: transcoding ten million source tiles cost about $1,500, while the non-spot instance path was estimated at about $50,000. The local Ceph system remains the working tier for development, rather than the global distribution tier.

Recommendation 6

Funders should budget for post-project replication and archival stewardship, with interoperable public repositories as a default route for long-lived datasets. *Addresses* R4

2.6 Common operational insights

Making application code portable is only one part of the puzzle. At this data volume, the surrounding scheduler, network, and storage components are equally important for optimal throughput. The following recurring findings condense these observations:

Distributed-stack failures. While the PyTorch model code was portable, multi-node execution was not. Dawn needed remedies for oneCCL transport and MPI launch failures; AMD systems needed MIOpen and graph-compilation workarounds; Isambard encountered NCCL hangs; and Zenith exposed RCCL topology disagreement. Each facility would benefit from a tested reference job covering its compiler, communication library, fabric, and topology.

Job shape and allocation. We sometimes required many tightly coupled GPUs for v2 teacher training, whereas the weather study comprised independent one-GPU runs. Providers should combine reserved scale-out windows, job arrays, and smaller inference jobs where isolation permits, while exposing device health and topology to the scheduler.

Remote ingestion repeatedly saturated a shared network source. Microsoft Planetary Computer throttled high-concurrency Vultr data fetches, and Zenith's NAT caused additional nodes to subdivide the same external per-IP request allowance. Isambard avoided this failure because compute nodes had public addresses. Clients capable of pulling 10 Gb/s or more can also make an institutional uplink the bottleneck; our Dawn and Zenith transfers, coordinated through a departmental host, triggered major network-use alerts through the Cambridge network monitoring. Data-intensive services need per-project egress identities, managed caching, transfer accounting, and an agreed institutional path for sustained external traffic.

Checkpointing did not provide predictable wall-clock completion. Device and node failures were recoverable on all three accelerator ecosystems, but recovery time was unpredictable. This was most costly on Isambard, where mandatory 24-hour segments could wait several days before resuming. Facilities supporting large training runs should offer reservations or priority continuation and report expected queue delay for the requested job shape.

Storage limits appeared at several layers. We often exhausted Isambard's inode allocation despite adequate "byte" capacity. Zenith initially failed similarly; after its inode allocation was increased, we staged multiple terabytes on Lustre and fully fed the GPUs. Providers should state capacity, inode, metadata-rate, purge, and object-count limits together, and provide an explicit path from temporary HPC storage to durable publication.

Supercomputer storage needs a public transition path. Domain repositories such as CEDA, the Environmental Information Data Centre (EIDC), and the British Oceanographic Data Centre (BODC) provide established stewardship for environmental observations [7, 39, 47]. UKRI's Digital Research Infrastructure programme is now developing shared repository and HPC data-curation services, but these do not yet provide a domain-neutral archive for large machine-learning models and embedding collections [49]. Projects therefore fall back to commercial clouds with short-term commitments or startups such as Hugging Face. HPC programmes should fund and operate open, replicated repositories that accept model artefacts and embeddings at supercomputer scale, with clear migration and preservation guarantees consistent with UKRI's open-data expectations [50]. Source Cooperative has solved our immediate distribution problem, but its philanthropic funding model needs replication and adoption by major public bodies, including the EU, to support durable open geospatial infrastructure [27, 44].

Batch logs were insufficient for performance diagnosis. Across platforms, diagnosing bottlenecks required accelerator counters (including memory-stall and utilisation metrics), correlated timings, node-health events, network volumes, and scheduler state. Their absence made it difficult to distinguish application faults from failed devices, fabric problems, or remote throttling. Providers should expose these counters and offer production-equivalent interactive developer nodes alongside batch queues.

HPC administrators provided effective technical support. Every facility team responded quickly to tickets and configuration questions, for which we are incredibly grateful. This was essential when faults crossed scheduler, interconnect, and node boundaries that users could not inspect. The Azure and Google Cloud support available to the project offered no comparable route to an infrastructure engineer, leaving data-access and rate-limit diagnosis slower and less conclusive.

Coding agents became the primary operational interface. At the start of the Dawn work, coding assistants were largely confined to generating fragments in a web interface that researchers copied into terminals. By the Zenith work, the team used coding agents almost exclusively for environment construction, Slurm scripting, log analysis, and performance diagnosis.

Successful agentic use favours command-line tools with structured output and complete, versioned documentation. An interesting consequence of this is that while Python and shell scripts remain the dominant languages, we also used less conventional languages like OCaml [8] and OxCaml [31] for high-throughput storage and serving components.

Recommendation 7

Publish facility-specific, least-privilege coding agent skills with structured scheduler, topology, quota and telemetry interfaces, sandboxing, confirmation for consequential actions, and retained audit logs. *Addresses* R6

3 Responsible compute and energy reporting

Tracking energy consistently was difficult because each provider exposed a different accounting boundary: some reported GPU-hours, some exposed only allocation totals, and cloud providers did not provide comparable job-level power records. A common API for device-hours, average and peak power, queue delay, failures, data movement, and facility carbon intensity would make responsible reporting reproducible across systems.

The following inline table reports known compute and reproducible power-cap bounds. For recorded device-hours H and power cap P, the bound is $E_{\max} = HP$. It assumes maximum power for every hour. It is therefore distinct from measured electricity. The figures exclude facility overhead, storage, and networking; Isambard is the exception in which the 660 W cap covers the combined Grace CPU and H100 GPU superchip [6]. Nested Isambard rows must not be summed.

Activity	Accounting and cap	Energy bound
Dawn v1	Records unavailable; 0.600 kW/GPU [26].	Needs Slurm records.
AMD cloud v1	About 6,200 GPU-hours; 0.750 kW/GPU [1, 21].	4.65 MWh.
Vultr v1/v1.1	Records unavailable; 1.000 kW/GPU [1].	Not derivable from tile count.
Isambard 2	800,000 GPU-hours; 0.660 kW/GH200 [6].	528 MWh.
V2 scaling	About 320,000 GPU-hours including failures; 0.660 kW/GH200.	211.2 MWh.
Weather paper	About 9,600 GPU-hours; 0.660 kW/GH200 [43].	6.34 MWh.
Zenith distillation	About 380 GPU-hours; 1.400 kW/GPU [3].	0.532 MWh.

The available records do not support a defensible CO2-equivalent total. That calculation requires job-level average power or energy, facility power usage effectiveness, and time-resolved grid carbon intensity. Future runs should preserve Slurm accounting, failed-job time, energy telemetry, bytes transferred, and successful output area. These data would support energy-per-checkpoint and energy-per-square-kilometre metrics.

Recommendation 8

Expose job-level energy, average power, queue delay, failure cause, and data-transfer counters. Publish facility power usage effectiveness and time-resolved carbon data where possible. *Addresses* R5

4 Conclusions and Future work

We are continuing to work on several new model iterations and architectures, including a change-tracking model using NVIDIA A10 instances on Microsoft Azure as well as Zenith's AMD cluster. Finding suitable GPUs in a large shared cloud has itself been an operational constraint; an A10 ECC watchdog was needed to detect unhealthy instances [10] in Azure.

We also plan to deploy the Intel AMX CPU inference path for large scale inference, which would make large embedding releases less dependent on scarce GPU allocations and easier to reproduce on reserved CPU-only resources [14]. Encouragingly, other community members such as dClimate and AWS Open Data have already reproduced and optimised our inference pipeline, and produced wall-to-wall embeddings of v1.1 that are fully compatible[2] with the Cambridge generated ones.

The Tessera project has involved a lot of GPU and CPU time and dozens of petabytes of data motion across multiple providers, countries, and organisations. The resulting models and embeddings are fully reproducible and extensible, and are now in use by many organisations. We are deeply grateful for the financial, technical, and cross-organisational support that made this work possible, and look forward to the coming years. If you find this useful or wish to exchange tips, then feel free to join our open-registration Zulip instance[3] and introduce yourself.

Acknowledgements

The Isambard-AI processing was supported by the UK Research and Innovation AI Research Resource under award ST/AIRR/I-A-I/1023. Isambard-AI is operated by the Bristol Centre for Supercomputing; we particularly thank Simon McIntosh-Smith and the Isambard team for support [37].

The authors acknowledge the use of resources provided by the Dawn and Zenith supercomputers. Zenith is operated by the University of Cambridge and is funded by the UK Government's Department for Business, Innovation, Science and Trade (BSIT) and UK Research and Innovation (UKRI) as part of the UK Artificial Intelligence Research Resource (AIRR) [ST/Z000890/1], and the National Compute Resource (NCR), delivered in partnership with Dell Technologies and AMD. We thank Ryan Daniels, Deepak Aggarwal,

[2] See https://registry.opendata.aws/tessera/
[3] See https://eeg.zulipchat.com

Christopher Edsall, Paul Calleja and Stuart Rankin at the University of Cambridge for their support.

We also thank Jane Street and Tarides for financial and machine donations; Kieran Mansley, Cathal McCabe, Rowan Pavlovic, Maggie Anderson, and Joseph Cowell at AMD; Kasia Hilborne, Kevin Cochrane, and AJ Karolczak at Vultr; Bogdan Stefan Pop at Intel, and Steve Davey and his team at NVIDIA. We acknowledge UKRI, STFC Durham DiRAC HPC Facility, the Microsoft AI for Good Lab, Google, Dr Robert Sansom, and John Bernstein. We thank the Dawn, AMD Developer Cloud, Vultr, Isambard-AI, and Zenith administrators for their responsive technical support, the open-data providers on which Tessera depends, and for the many members of the wider Tessera and Cambridge Conservation Initiative community who have contributed feedback and insights.

The storage provided by Source Coop was made possible by Jed Sundwall from Radiant Earth and Isaac Corley from Taylor Geospatial, to whom we are most grateful for timely assistance at a time when every SSD in Cambridge seemed to be stuffed with embeddings.

References

[1] Advanced Micro Devices, Inc. 2025. AMD CDNA 3 Architecture White Paper. The published maximum powers are 750 W for MI300X and 1000 W for MI325X; accessed 14 September 2026. https://www.amd.com/content/dam/amd/en/documents/instinct-tech-docs/white-papers/amd-cdna-3-white-paper.pdf

[2] Advanced Micro Devices, Inc. 2026. AMD Developer Cloud. Accessed 14 September 2026. https://www.amd.com/en/developer/resources/cloud-access/amd-developer-cloud.html

[3] Advanced Micro Devices, Inc. 2026. AMD Instinct MI355X GPU Specifications. The published typical board power is 1400 W; accessed 14 September 2026. https://www.amd.com/en/products/accelerators/instinct/mi350/mi355x.html

[4] Anna Allen, Stratis Markou, Will Tebbutt, James Requeima, Wessel P. Bruinsma, Tom R. Andersson, Michael Herzog, Nicholas D. Lane, Matthew Chantry, J. Scott Hosking, and Richard E. Turner. 2025. End-to-end data-driven weather prediction. *Nature* 641, 8065 (2025), 1172–1179. doi:10.1038/s41586-025-08897-0

[5] James G. C. Ball, Jana Annika Wicklein, Zhengpeng Feng, Jovana Knezevic, Sadiq Jaffer, Anil Madhavapeddy, Clement Atzberger, Michele Dalponte, and David A. Coomes. 2026. Geospatial Foundation Models Enable Data-efficient Tree Species Mapping in Temperate Mountain Forests. *Science of Remote Sensing* (2026). doi:10.1016/j.srs.2026.100466

[6] Bristol Centre for Supercomputing. 2026. Isambard System Specifications. Accessed 14 September 2026. https://docs.isambard.ac.uk/specs/

[7] Centre for Environmental Data Analysis. 2026. CEDA environmental data archive. Accessed 16 September 2026. https://www.ceda.ac.uk/

[8] Stephen Dolan, Leo White, and Anil Madhavapeddy. 2014. Multicore OCaml. In *OCaml Workshop*, Vol. 2.

[9] Mark Elvers. 2025. Ceph Placement Groups. Technical note. 9 December 2025; accessed 14 September 2026. https://www.tunbury.org/2025/12/09/ceph-placement-groups/

[10] Mark Elvers. 2026. Azure A10 ECC watchdog. Weeknote. 4 June 2026. https://www.tunbury.org/2026/06/04/azure-a10-ecc-watchdog/

[11] Mark Elvers. 2026. Ceph Notes. Technical note. 6 January 2026; accessed 14 September 2026. https://www.tunbury.org/2026/01/06/ceph-notes/

[12] Mark Elvers. 2026. Expanding the Ceph Cluster to 600 TB. Technical note. 27 March 2026; accessed 14 September 2026. https://www.tunbury.org/2026/03/27/ceph-expansion/

[13] Mark Elvers. 2026. Tessera Pipeline. Technical note. 25 February 2026. https://www.tunbury.org/2026/02/25/teserra-pipeline/

[14] Mark Elvers. 2026. Tessera v1.1 inference: faster per-FLOP, slower per-tile. Technical note. 29 April 2026; accessed 16 September 2026. https://www.tunbury.org/2026/04/29/tessera-v1-1-benchmarks/

[15] Mark Elvers. 2026. Week 28, 2026. Weeknote. 13 July 2026. https://www.tunbury.org/2026/07/13/week-28/

[16] Mark Elvers. 2026. Week 31, 2026. Weeknote. 3 August 2026. https://www.tunbury.org/2026/08/03/week-31/

[17] Mark Elvers. 2026. Week 34, 2026. Weeknote. 24 August 2026. https://www.tunbury.org/2026/08/24/week-34/

[18] T. Erwin and J. Sweetkind-Singer. 2009. The National Geospatial Digital Archive: A Collaborative Project to Archive Geospatial Data. *Journal of Map and Geography Libraries* 6, 1 (2009), 6–25. doi:10.1080/15420350903432440

[19] Zhengpeng Feng. 2025. *AMD Experience: Porting TESSERA from Dawn to MI300X.* Technical Report. University of Cambridge. Internal technical report, with author clarification of 24 March 2025.

[20] Zhengpeng Feng, Clement Atzberger, Sadiq Jaffer, Jovana Knezevic, Silja Sormunen, Robin Young, Madeline Lisaius, Markus Immitzer, Toby Jackson, James Ball, David A. Coomes, Anil Madhavapeddy, Andrew Blake, and Srinivasan Keshav. 2026. Applications of the TESSERA Geospatial Foundation Model to Diverse Environmental Mapping Tasks. *SSRN Electronic Journal* (2026). doi:10.2139/ssrn.6142416

[21] Zhengpeng Feng, Clement Atzberger, Sadiq Jaffer, Jovana Knezevic, Silja Sormunen, Robin Young, Madeline C. Lisaius, Markus Immitzer, Toby Jackson, James Ball, David A. Coomes, Anil Madhavapeddy, Andrew Blake, and Srinivasan Keshav. 2026. TESSERA: Temporal Embeddings of Surface Spectra for Earth Representation and Analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 34818–34831. https://openaccess.thecvf.com/content/CVPR2026/html/Feng_TESSERA_Temporal_Embeddings_of_Surface_Spectra_for_Earth_Representation_and_CVPR_2026_paper.html

[22] Zhengpeng Feng, Sadiq Jaffer, Ira Shokar, Jovana Knezevic, James Ball, Pedro Sousa, Mark Elvers, Madeline Lisaius, Clement Atzberger, Robin Young, Aneesh Naik, Niall Robinson, David Coomes, Anil Madhavapeddy, and Srinivasan Keshav. 2026. TESSERA v2: Scaling Pixel-wise Earth Foundation Models. *arXiv preprint arXiv:2607.03949* (2026). doi:10.48550/arXiv.2607.03949

[23] GeoTESSERA project. 2026. GeoTESSERA: Open Earth-observation embeddings. Project website. Accessed 14 September 2026. https://geotessera.org/

[24] GeoTessera project. 2026. GeoTessera v0.10.0. Zenodo software archive. Created 26 August 2026; accessed 15 September 2026. doi:10.5281/zenodo.22108692

[25] GeoTessera project. 2026. Tessera model weights. Hugging Face Hub. Accessed 15 September 2026. https://huggingface.co/geotessera

[26] Intel Corporation. 2026. Intel Data Center GPU Max 1550 Specifications. The published thermal design power is 600 W; accessed 14 September 2026. https://www.intel.com/content/www/us/en/products/sku/232873/intel-data-center-gpu-max-1550/specifications.html

[27] Alex Leith et al. 2024. Cloud Native Geospatial: Realizing the Digital Earth Vision. In *2024 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 6886–6889. doi:10.1109/IGARSS53475.2024.10642711

[28] Madeline C. Lisaius, Andrew Blake, Clement Atzberger, and Srinivasan Keshav. 2026. Towards Improved Crop Type Classification: A Compact Embedding Approach Suitable for Small Fields. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2026), 873–880. doi:10.5194/isprs-annals-XI-2-2026-873-2026

[29] Madeline C. Lisaius, Andrew Blake, Srinivasan Keshav, and Clement Atzberger. 2024. 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 (2024), 13162–13168. doi:10.1109/JSTARS.2024.3426044

[30] Madeline C. Lisaius, Srinivasan Keshav, Andrew Blake, and Clement Atzberger. 2026. Embedding-based Crop Type Classification in the Groundnut Basin of Senegal. *arXiv preprint arXiv:2601.16900* (2026). doi:10.48550/arXiv.2601.16900

[31] Anton Lorenzen, Leo White, Stephen Dolan, Richard A. Eisenberg, and Sam Lindley. 2024. Oxidizing OCaml with Modal Memory Management. *Proceedings of the ACM on Programming Languages* 8, ICFP (2024). doi:10.1145/3674642

[32] Anil Madhavapeddy. 2026. Celebrating a year of TESSERA embeddings and releasing GeoTESSERA 0.10. Technical note. 27 August 2026; accessed 14 September 2026. https://anil.recoil.org/notes/geotessera-a-year-on

[33] Anil Madhavapeddy. 2026. Streaming millions of TESSERA tiles over HTTP with Zarr v3. Technical note. 14 March 2026; accessed 14 September 2026. doi:10.59350/tk0er-ycs46

[34] Anil Madhavapeddy. 2026. Tessera v1.1 released, with smoother and temporally stable embeddings. Technical note. 12 June 2026; accessed 15 September 2026. doi:10.59350/vcqjp-24y05

[35] Anil Madhavapeddy, David J. Scott, and Justin Cormack. 2026. A Decade of Docker Containers. *Commun. ACM* 69, 3 (2026), 56–65. doi:10.1145/3761803

[36] Karen Majewicz. 2026. Preserving Today's Public Geospatial Data for Future Researchers: A Comparative Analysis of International Approaches. University Digital Conservancy. Accessed 15 September 2026. https://hdl.handle.net/11299/279588

[37] Simon McIntosh-Smith, Sadaf Alam, and Christopher Woods. 2025. Isambard-AI: a leadership-class supercomputer optimised specifically for Artificial Intelligence. In *CUG 2024: Proceedings of the Cray User Group*. Association for Computing Machinery, 44–54. doi:10.1145/3725789.3725794

[38] Microsoft. 2026. Microsoft Planetary Computer STAC API. Technical documentation. Accessed 15 September 2026. https://planetarycomputer.microsoft.com/docs/reference/stac/

[39] National Oceanography Centre. 2026. British Oceanographic Data Centre. Accessed 16 September 2026. https://www.bodc.ac.uk/

[40] Open Geospatial Consortium. 2025. SpatioTemporal Asset Catalog (STAC). OGC Community Standard. Accessed 15 September 2026. https://stacspec.org/en/

[41] PyTorch Contributors. 2024. ROCm Inductor: `torch.compile` `assert_size_stride` Assertion Error. GitHub issue 137414. Opened 6 October 2024; accessed 14 September 2026. https://github.com/pytorch/pytorch/issues/137414#issuecomment-2412600401

[42] Microsoft Open Source, Matt McFarland, Rob Emanuele, Dan Morris, and Tom Augspurger. 2022. *microsoft/PlanetaryComputer: October 2022.* doi:10.5281/zenodo.7261897

[43] Pedro Sousa, Will Tebbutt, Sadiq Jaffer, Robin Young, Anil Madhavapeddy, and Richard E. Turner. 2026. Earth observation embeddings are effective sub-grid descriptors for probabilistic weather downscaling. *arXiv preprint arXiv:2608.12271* (2026). doi:10.48550/arXiv.2608.12271

[44] Jed Sundwall. 2025. *Emergent Standards: Enabling Collaboration Across Institutions.* Technical Report. The Institutional Architecture Lab (TIAL). https://tial.org/wp-content/uploads/2025/11/TIAL-Emergent-Standards-V18112025.pdf

[45] TrendForce. 2025. NAND Flash Wafer Supply Tightens Further, Some Products See Over 60% Contract Price Hikes in November. 1 December 2025; accessed 14 September 2026. https://www.trendforce.com/presscenter/news/20251201-12807.html

[46] TrendForce. 2026. Memory Price Outlook for 1Q26 Sharply Upgraded. 2 February 2026; accessed 14 September 2026. https://www.trendforce.com/presscenter/news/20260202-12911.html

[47] UK Centre for Ecology and Hydrology. 2026. Environmental Information Data Centre. Accessed 16 September 2026. https://eidc.ac.uk/

[48] UK Research and Innovation. 2025. AIRR compute opportunity: AI for Science. Funding opportunity. Published 21 November 2025; accessed 15 September 2026. https://www.ukri.org/opportunity/airr-compute-opportunity-ai-for-science/

[49] UK Research and Innovation. 2026. Digital Research Infrastructure Programme. Accessed 16 September 2026. https://www.ukri.org/what-we-do/browse-our-areas-of-investment-and-support/digital-research-infrastructure/

[50] UK Research and Innovation. 2026. Making your research data open. Updated 15 September 2026; accessed 16 September 2026. https://www.ukri.org/manage-your-award/publishing-your-research-findings/making-your-research-data-open/

[51] University of Cambridge Earth Observation group. 2026. TESSERA source code and model resources. GitHub repository. Accessed 14 September 2026. https://github.com/ucam-eo/tessera

[52] University of Cambridge Research Computing Services. 2026. Dawn Intel Data Center GPU Max User Guide. Accessed 14 September 2026. https://docs.hpc.cam.ac.uk/hpc/user-guide/pvc.html

[53] University of Cambridge Research Computing Services. 2026. Zenith Quick Start Guide. Accessed 14 September 2026. https://docs.hpc.cam.ac.uk/user-guide/zenith-quickstart.html

[54] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Carlos Maltzahn. 2006. Ceph: A Scalable, High-Performance Distributed File System. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06).* USENIX Association, 307–320. https://www.usenix.org/events/osdi06/tech/full_papers/weil/weil_html

[55] Robin Young and Srinivasan Keshav. 2026. Interpolation of GEDI Biomass Estimates with Calibrated Uncertainty Quantification. *arXiv preprint arXiv:2601.16834* (2026). doi:10.48550/arXiv.2601.16834

[56] Robin Young and Srinivasan Keshav. 2026. Spatiotemporal Interpolation of GEDI Biomass with Calibrated Uncertainty. *arXiv preprint arXiv:2604.03874* (2026). doi:10.48550/arXiv.2604.03874

[57] Robin Young, Michael E. Van Nuland, E. Toby Kiers, Tomáš Větrovský, Petr Kohout, Petr Baldrian, and Srinivasan Keshav. 2026. Below-ground Fungal Biodiversity Can be Monitored Using Self-Supervised Learning Satellite Features. *arXiv preprint arXiv:2604.09818* (2026). doi:10.48550/arXiv.2604.09818

[58] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania, Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. 2023. PyTorch FSDP: Experiences on Scaling Fully Sharded Data Parallel. *arXiv preprint arXiv:2304.11277* (2023). doi:10.48550/arXiv.2304.11277