

Odeai: Observation-Driven Weather Forecasting Through End-to-End Differentiable Systems

Odeai

April 20, 2026

Abstract

Current numerical weather prediction relies on a pipeline of data assimilation and physical forecasting whose individual components have well-known limitations: Gaussian error assumptions in variational assimilation, poorly controlled physical parameterizations, and a critical coupling between observation operators and model physics. Recent AI approaches have achieved impressive forecast skill but remain dependent on reanalyses that inherit these biases. Meanwhile, observation-based models such as MetNet 3 (Andrychowicz et al., 2023) and OMG-HD (Zhong et al., 2024) have demonstrated that raw observations can drive accurate short-range predictions, suggesting a viable alternative path. This white paper presents the Odeai vision: an observation-centric, end-to-end differentiable forecasting system that combines learned observation embeddings with a lightweight dynamical core, aiming to operate in a closed loop free from reanalysis dependency.

Contents

1	Background on Weather Prediction	3
2	Related Works and Limitations	3
2.1	Numerical Weather Prediction	3
2.1.1	Dynamical Core	4
2.1.2	Physical Parameterization	4
2.1.3	Data Assimilation in Practice	4
2.1.4	Assimilation Latency	5
2.2	AI-Enabled Approaches	5
2.2.1	Reanalysis-Trained Models	5
2.2.2	Hybrid Approaches	6
2.2.3	Observation-Based Approaches	6
3	Odeai	6
3.1	Core Claims	7
3.2	System Components	7
3.3	Validation and Evaluation	8
3.4	Opportunities	8
4	Implementation and Technical Aspects	9
4.1	Type of Learning	9
4.2	Neural Network Architecture	9
4.3	Software	10
4.4	Data Volume	10
4.5	Data Types	10
4.6	Compute Duration Estimate	10

4.6.1	Reference Baseline	10
4.6.2	Scaling to Target Models	11
4.6.3	Budget Summary	12
4.6.4	Resource Consumption Schedule	12
4.7	Licensing, Intellectual Property, and Data Confidentiality	12

1 Background on Weather Prediction

Modern weather prediction rests on three interacting components: a data assimilation system that estimates the current atmospheric state from observations, a dynamical core that advances this state forward in time using the equations of fluid dynamics, and a set of physical parameterizations that approximate processes too small or complex to be explicitly resolved. We briefly introduce each component and how they interact in the operational forecasting cycle.

Data assimilation (DA). Data assimilation is the process of optimally combining a prior forecast (the *background*) with new observations to estimate the current atmospheric state (the *analysis*). It can be framed as an inverse problem: a high-dimensional hidden state must be inferred from sparse, indirect, and heterogeneous measurements. The challenge is compounded by the fact that observations are unevenly distributed in space and time, and many of them (e.g. satellite radiances) do not measure model variables directly.

Dynamical core. The dynamical core is the numerical solver for the governing equations of atmospheric fluid dynamics (Navier–Stokes on a rotating sphere, under hydrostatic or non-hydrostatic approximation). It handles *resolved* scales, i.e. motions that the computational grid can explicitly represent, and advances the prognostic variables (wind components, temperature, pressure, and humidity) forward in time. Because these equations derive from fundamental conservation laws, the dynamical core requires very few tunable parameters.

Physical parameterization. Physical parameterizations are approximate models for processes occurring below the grid resolution (*sub-grid* processes): convection, turbulence, radiation transfer, cloud microphysics, and surface exchanges. Unlike the dynamical core, parameterizations are semi-empirical and contain tunable parameters whose calibration strongly influences forecast quality.

The operational weather prediction cycle can be summarized as follows:

$$\begin{array}{c} (X_{\text{obs}})_t \text{ and } (X_{\text{prior}})_t \xrightarrow{\text{DA}} (X_{\text{analysis}})_t \\ (X_{\text{analysis}})_t \xrightarrow[\text{Physics of unresolved interactions}]{\text{Dynamical core}} (X_{\text{forecast}})_{t+1} \end{array}$$

At time t , observations X_{obs} and a prior forecast X_{prior} are combined via data assimilation (DA) to produce an analysis X_{analysis} . This analysis serves as the initial condition for the forecast model, which integrates the dynamical core and physical parameterizations forward in time to produce a forecast at $t+1$.

The forecast at $t+1$ then serves as the prior for the next assimilation cycle, forming a closed loop.

2 Related Works and Limitations

We now review the main approaches to weather forecasting — traditional numerical weather prediction and recent AI-enabled methods — highlighting the strengths and limitations of each. This analysis motivates the design choices presented in Section 3.

2.1 Numerical Weather Prediction

Numerical weather prediction (NWP) systems combine two main components: a dynamical core that solves the equations of atmospheric motion on a discrete grid, and physical parameterizations that represent sub-grid processes not captured by those equations. We discuss each in turn, followed by the data assimilation framework that provides initial conditions.

2.1.1 Dynamical Core

The dynamical core solves simplified forms of the Navier–Stokes equations, enforcing conservation of mass, conservation of energy, thermodynamic relations, and transport of momentum and tracers. From these equations the model computes the prognostic variables (wind, temperature, pressure, humidity) at each grid point and time step.

Because the dynamical core is rooted in fundamental conservation laws, it provides a coherent evolution at large scales, requires very few adjustments and no tunable parameters, and can be extrapolated to weather regimes not present in the training data.

Limitations

- Computationally expensive at high resolution: halving the grid spacing roughly multiplies the cost by a factor of 8–16 (three spatial dimensions plus shorter time steps).
- Still requires parameterization for all processes below the resolved scale, so the dynamical core alone does not constitute a complete forecast model.

2.1.2 Physical Parameterization

Physical parameterization is the main source of forecast error, alongside the error on the initial condition. Many sub-grid phenomena are only partially resolved at operational resolutions: convection, turbulence, and cloud microphysics. The underlying physics is sometimes inaccessible at the model resolution or inherently difficult to represent.

Limitations

- Main source of forecast error alongside initial condition error.
- Calibration is critical yet poorly controlled; introducing new processes may break existing calibrations.
- Can dominate the computational cost in some models.

2.1.3 Data Assimilation in Practice

Data assimilation seeks the analysis state X_a that best reconciles the prior forecast with available observations. In variational formulations, this is achieved by minimizing a cost function that balances fidelity to the background state and fidelity to the observations. The dominant operational method, four-dimensional variational assimilation (4D-Var), minimizes:

$$J(X_a) = \underbrace{(X_a - X_b)^T B^{-1} (X_a - X_b)}_{\text{background term}} + \underbrace{\sum_t (H(x_t) - y_t)^T R^{-1} (H(x_t) - y_t)}_{\text{observation term}} \quad (1)$$

where X_b is the background (prior) state, B is the background error covariance matrix, H is the observation operator mapping the model state to observation space, y_t are the observations at time t , and R is the observation error covariance matrix. The interpretation is as follows:

- Term 1: the analysis should remain close to the prior forecast.
- Term 2: the analysis, mapped through the observation operator, should remain close to the actual observations.

These errors propagate into data assimilation through the observation operator H . When observations are not simple interpolations of model fields (for instance, satellite radiances requiring a full radiative transfer model), two compounding error sources arise:

- a systematic bias in H itself, inherited from the parameterization,
- a state-dependent error $H(x_{\text{true}}) - H(x)$, amplified by the imprecision of the physical model.

This critical coupling between parameterization and assimilation limits the effective use of many observation types, particularly those most sensitive to atmospheric composition and cloud structure.

Limitations

- Gaussianity assumption on errors: $x_a - x_b \sim \mathcal{N}(0, B)$ and $y_t - H(x_t) \sim \mathcal{N}(0, R)$. This does not hold for many phenomena: clouds, deep convection, precipitation, radar. The assumption is especially problematic during extreme events, precisely the situations where forecast accuracy matters most: precipitation is bounded and highly skewed, convective errors are often bimodal (system present vs. absent), and both background and observation error magnitudes can far exceed the climatological statistics encoded in B and R .
- Observation operators can be sub-optimal, especially for non-trivial measurements (satellite radiances, radiative transfer).
- Critical coupling between parameterization biases and H : errors in the physical model contaminate the assimilation.

2.1.4 Assimilation Latency

Typically, 4D-Var assimilation is performed over a 6-hour window and takes about one hour of computation. The end-to-end latency to produce an analysis is often around 3 hours, using data that is at best 2 hours old. This constrains the timeliness of operational forecasts and limits the exploitation of rapidly updating observation sources.

By contrast, an AI-based system coupled with a lightweight dynamical core can ingest observations and produce updated forecasts in minutes rather than hours, enabling near-real-time assimilation cycles and faster response to rapidly evolving weather events.

2.2 AI-Enabled Approaches

2.2.1 Reanalysis-Trained Models

Most current AI weather models focus on replacing the dynamical core and physical parameterizations with a deep learning model trained on reanalysis archives. Notable examples include GraphCast (Lam et al., 2023), Pangu-Weather (Bi et al., 2023), FourCastNet (Pathak et al., 2022), and GenCast (Price et al., 2025). These approaches have demonstrated strong predictive capabilities, achieving competitive RMSE and anomaly correlation scores at medium range (3–10 days) compared to operational NWP systems. However, they remain fully dependent on high-quality data assimilation for both their training data and their initial conditions at inference time.

In many cases, the loss function used to train these models is a variant of MSE. As a result, the model tends to predict the conditional expectation, which has a smoothing effect on forecasts that worsens with the forecast horizon, mechanically decreasing predicted intensity. Solutions have begun to address this issue: generative models such as diffusion models; CRPS-derived losses combined with perturbation systems; and hybrid models whose physical constraints and design (e.g. ML limited to the vertical column) strongly limit the smoothing effect. These remain active areas of research.

2.2.2 Hybrid Approaches

Hybrid approaches aim to combine the physical grounding of NWP with the flexibility of machine learning. NeuralGCM (Kochkov et al., 2024), for instance, retains a differentiable dynamical core while replacing traditional parameterizations with learned neural-network components. This preserves large-scale physical consistency and allows end-to-end training. However, NeuralGCM is still trained on reanalysis data and therefore inherits the biases present in those archives.

2.2.3 Observation-Based Approaches

Another family of AI approaches has shown that it is possible to partially bypass analyses and reanalyses for forecasting. MetNet 3 (Andrychowicz et al., 2023) and OMG-HD (Zhong et al., 2024) use only observations as input to their models and have demonstrated that, for many variables, very short-term forecasts produced this way are more accurate than the analysis or reanalysis itself.

This finding is significant for two reasons. First, it demonstrates that **a neural network can learn the mapping from raw, heterogeneous observations to atmospheric state variables** without relying on the hand-crafted observation operators and error models of traditional data assimilation. In doing so, the network implicitly learns to handle the irregularity, sparsity, and indirectness of real-world measurements — challenges that conventional DA addresses through explicit statistical modelling (covariance matrices B and R , observation operator H) with all its attendant limitations. Second, and more fundamentally, the fact that observation-based forecasts can *outperform* the analysis itself at short lead times indicates that the conventional DA pipeline actively discards or distorts information present in the observations. The Gaussian assumptions, the coupling to imperfect physical models, and the latency of the assimilation window all contribute to an information loss that a learned system can partially recover. In other words, the quality ceiling imposed by reanalyses on downstream AI models is not a fundamental limit of the observations — it is an artefact of the processing pipeline.

However, these observation-based methods are most skillful at short lead times; the gap with traditional NWP closes beyond approximately +24 h. This degradation may be a sign that purely data-driven models lack the physical priors (conservation laws, large-scale dynamics) that NWP exploits, and that some form of physical preconception is necessary to maintain coherence at longer horizons.

3 Odeai

Limitations of current weather prediction systems all point to a common structural issue: observations, physical dynamics, and uncertainty quantification are treated as separate, loosely coupled stages. This suggests that a system designed from the ground up around observations could retain the benefits of learned dynamics while sidestepping the limitations of reanalysis dependency. This is the path that Odeai pursues.

Limitations of Current Systems

- **Reanalysis dependency.** For fields heavily dependent on physical parameterization, reanalyses are largely improvable. AI models trained on these archives reproduce their biases, limiting the effective ceiling of forecast skill.
- **Blurry forecasts from pixel-level MSE.** When MSE is applied in variable or pixel space, the model learns to predict the conditional mean, producing increasingly blurry forecasts as the lead time grows.
- **Short-range ceiling of observation-based models.** Skill advantage over NWP and reanalysis-trained models is limited to short lead times (approximately < 24 h); beyond this horizon the gap closes rapidly. The absence of explicit physical constraints (conservation laws, large-scale dynamics) may explain the loss of coherence at longer ranges.

Odeai is designed to address such limitations by building an end-to-end differentiable system where observations drive every component, physical priors enforce coherence at longer horizons, and ensemble generation accounts for the chaotic nature of the atmosphere from the outset. We now describe the core principles and components of this system.

3.1 Core Claims

Our Claims at Odeai

- Observations are under-exploited and should play a central role in forecasting and in the training of the various forecasting components. The medium- to long-term goal is to master data assimilation from raw observations, **removing the dependency on third-party analyses and reanalyses** such as ERA5, and to operate the forecasting system in a closed loop.
- Integrating physics through differentiable solvers — even greatly simplified ones — can enforce large-scale coherence and **provide the physical priors needed to maintain forecast skill beyond the short range**. This is particularly relevant for temporal dynamics, where conservation laws and advective transport provide strong inductive biases.
- Weather is inherently chaotic: **ensemble forecasting must be a first-class design goal**, not an afterthought. Producing calibrated probabilistic predictions is essential both for decision-making and for informing the assimilation cycle through ensemble-based uncertainty estimates.
- Data assimilation can reduce the observational history required for skillful prediction, by distilling the relevant information from past observations into a compact, physically consistent state representation.

3.2 System Components

- **Joint observation embedding.** Train a shared encoder that maps heterogeneous observation sources (satellite imagery, radar reflectivities, point measurements) into a common latent space. This densifies the sparse, irregular observation coverage into a continuous latent state, enabling the model to learn the joint distribution of atmospheric variables across sensors. Self-supervised objectives such as the LeJEPa (Balestrieri and LeCun, 2025) loss can drive this training without requiring analysis labels.

- **JEPA-style forecast module.** Complement the encoder with a predictor operating in latent space, following the Joint Embedding Predictive Architecture (Assran et al., 2023; Balestrieri and LeCun, 2025). Rather than reconstructing raw observations at each step, the model predicts future latent states, avoiding pixel-level losses that lead to blurring. This can be augmented with an explicit transport module in latent space to capture advective dynamics and improve coherence at longer lead times.
- **Differentiable dynamical core.** Integrate a lightweight, differentiable solver for advection and large-scale dynamics into the forecasting pipeline. Even a simplified semi-Lagrangian transport scheme, when made differentiable, can enforce conservation properties and provide physically grounded temporal propagation, complementing the learned components and extending skillful forecast range beyond what purely data-driven models achieve.
- **Differentiable assimilation module.** Fine-tune the full system with an end-to-end differentiable assimilation loop: at each cycle, the model ingests new observations, updates its latent state, and propagates forward. By backpropagating through the assimilation step, the encoder and predictor learn to produce states that are both physically consistent and well-suited for observation ingestion.
- **Ensemble generation and probabilistic forecasting.** The ability to generate multiple forecast scenarios should be prioritized from the outset. Probabilistic forecasts are essential for decision-making in high-impact weather situations and enable ensemble-based assimilation, where the spread of the ensemble informs the uncertainty of the latent state and guides the weighting of new observations.

3.3 Validation and Evaluation

- **Direct observation verification.** Whenever possible, forecasts will be evaluated directly against observations rather than against analyses or reanalyses. This avoids circular evaluation where biases in the reference dataset mask real forecast errors, and ensures that the metrics reflect genuine predictive skill.
- **Focus on rare and high-impact events.** Standard aggregate scores (RMSE, ACC) tend to underweight the tail of the distribution. Special attention will be given to the evaluation on rare events (severe convection, heavy precipitation, rapid cyclogenesis) using threshold-based metrics (CSI, FSS), case studies, and conditional verification. This aligns with end-user needs, where forecast value is disproportionately concentrated in extreme events.
- **Probabilistic skill.** Ensemble forecasts will be assessed through proper scoring rules such as CRPS and reliability diagrams, ensuring that the system produces calibrated uncertainty estimates alongside sharp predictions.

3.4 Opportunities

Several converging developments make observation-centric forecasting tractable today:

- **Breakthroughs in self-supervised vision models.** Architectures such as JEPA (Assran et al., 2023), V-JEPA (Bardes et al., 2024), and LeJEPA (Balestrieri and LeCun, 2025) have shown that rich, structured representations can be learned from unlabeled data with strong theoretical and empirical foundations. These methods are directly applicable to satellite imagery and other gridded observation sources, enabling the construction of dense latent states from sparse, heterogeneous inputs without requiring paired analysis labels.

- **Demonstrated feasibility of observation-to-observation models.** MetNet 3 and OMG-HD have proven that neural networks can produce skillful forecasts directly from observations, bypassing the traditional analysis step. This validates the core premise that reanalyses are not a necessary intermediary.
- **Growing availability and diversity of observations.** The volume and coverage of weather observations continue to increase, spanning geostationary and polar-orbiting satellite imagery, microwave sounders, ground-based radar networks, surface weather stations, radiosondes, GNSS radio-occultation, and aircraft reports. A system designed to ingest observations natively is well positioned to exploit this expanding data landscape.
- **Open data policies.** National and international meteorological agencies are increasingly adopting open data policies, making operational observation archives accessible to private companies and research groups alike. EUMETSAT, NOAA, and the Copernicus Climate Data Store now provide free access to vast archives of satellite, radar, and in-situ data. This shift dramatically lowers the barrier to entry for building observation-centric systems and ensures that a growing share of the global observation network is available for model training and evaluation.

4 Implementation and Technical Aspects

This section provides details on the current implementation, data requirements, and compute resource estimates for the first phase of the Odeai project. While Section 3 describes the long-term vision, the work proposed here focuses on training continental-scale forecast models for precipitation, cloud cover, and solar radiation at 2 km resolution — building the observational data pipelines, model architectures, and training infrastructure that will underpin the full system.

4.1 Type of Learning

Our project leverages **self-supervised learning** rooted in the Joint-Embedding Predictive Architecture (JEPA) framework. A common self-supervised strategy is to reconstruct masked inputs pixel-by-pixel; however, this forces models to reproduce sensor-specific noise rather than meaningful atmospheric structure. Instead, our models learn *structured representations* of atmospheric state by predicting the latent embedding of one observational view (e.g. radar-derived precipitation, satellite-derived cloud cover) from another (e.g. multispectral satellite imagery, surface station measurements). This cross-modal prediction objective requires no explicit human-labeled ground truth, yet forces the encoder to learn representations that generalize across observational modalities. It is especially well-suited to data assimilation, where the core task is to fuse incomplete, noisy observations into a coherent atmospheric state; for forecasting, **autoregressive rollout training** extends the same framework by having the model iteratively predict future states from its own outputs.

4.2 Neural Network Architecture

The architecture is an **encoder–decoder (U-Net) with hybrid attention-convolution blocks**, combining efficient multi-axis self-attention (MaxViT) for global receptive field coverage with modern convolutional blocks (ConvNeXtV2) at each spatial scale. The **encoder is pre-trained** via the self-supervised JEPA objective described above, learning general-purpose atmospheric representations across observational modalities. Task-specific decoders are then attached for each downstream target; decoder layers may be partially shared across related targets (e.g. precipitation and cloud cover) or fully specialized, depending on the degree of physical coupling between variables. The model supports gradient checkpointing for memory-efficient training of high-resolution inputs.

Each model comprises approximately **20–100 M trainable parameters**, varying with the number of input channels and the chosen embedding dimension in the encoder. This range is consistent with current state-of-the-art vision architectures, and the hierarchical U-Net achieves spatial efficiency through progressive downsampling and multi-scale skip connections rather than operating on the full grid at a single resolution.

4.3 Software

The project relies on PyTorch as the core deep learning framework, complemented by standard open-source scientific and I/O Python libraries (rasterio, GDAL, pygrib, h5py). All domain-specific code (data pipelines, model architectures, training loops) is developed internally. No restrictive third-party license constraints apply. As we like to manage our dependency tree using Spack, we would appreciate some guidance on where dependencies should be compiled to avoid hitting storage quotas.

4.4 Data Volume

The dataset is constructed from multiple observational sources remapped onto regular grids. Data is split temporally: within each 10-day window, 7 days are used for training, 2 for testing, and 1 for evaluation.

Source	Temporal res.	Channels	Raw size/month
Geostationary satellite (IR/VIS)	10 min	11	~120 GB
Weather radar composites	2–10 min	3	~40 GB
Surface reanalysis	1 hour	~8	~10 GB
NWP analysis fields	1 hour	~54	~20 GB
Weather station observations	1 min	5–7	~4 GB
Polar satellite soundings	~12 hour	22	~2.5 GB
Static land features	—	~9	~0.5 GB

Table 1: Data sources for a continental domain at their native resolution.

For the existing 3-year (2021–2023) continental dataset, the data volume is significant: **~7–10 TB** on disk. Scaling to 6 years over continental-to-global domains, the estimated dataset size is **50–80 TB**. This range corresponds to the storage capacity that would give us the perfect conditions to experiment. We would appreciate it if you could get back to us if this quota is too high, so that we can consider a workaround. The train/test/evaluation split is approximately 70%/20%/10% by temporal sampling.

4.5 Data Types

The data consists of multi-channel gridded geophysical fields (structured 2D raster data): geostationary satellite radiance maps, weather radar reflectivity mosaics, NWP and reanalysis surface fields, sparse weather station observations projected onto regular grids on the fly, polar satellite microwave soundings, and static land features (topography, land/sea mask, coordinate encodings). Additionally, top-of-atmosphere solar radiation is computed on-the-fly as a geometric forcing input. All data is represented internally as multi-channel 2D tensors as input to the model.

4.6 Compute Duration Estimate

4.6.1 Reference Baseline

Our estimation is anchored on a measured internal benchmark:

Reference run: Training a single-step forecast model (10 min timestep, 1 output step) on a continental domain (CONUS) at 4 km resolution ($\sim 1799 \times 1059$ grid) takes 48 hours on a single NVIDIA RTX 4090.

Converting to H100 GPU-hours: given the $\sim 2\times$ effective throughput advantage of the H100 over the 4090 for mixed-precision training workloads of this type (accounting for both compute and memory bandwidth), this baseline corresponds to approximately 24 GPU-hours on one H100 device.

4.6.2 Scaling to Target Models

We target three models at 2 km resolution on continental domains, with exploratory work toward global coverage:

Model	Domain	Grid size (approx.)	Channels in/out
Precipitation	Continental	$\sim 3600 \times 2100$	~ 30 / 2–4
Cloud cover	Continental	$\sim 3600 \times 2100$	~ 25 / 3–5
Solar radiation	Continental	$\sim 3600 \times 2100$	~ 20 / 1–3

Table 2: Target model specifications at 2 km resolution.

The following scaling factors are applied cumulatively from the 24 H100-hour baseline:

1. **Resolution scaling (4 km $\rightarrow$ 2 km):** $4\times$ more grid points. Due to the hierarchical U-Net architecture, compute scales sub-linearly, yielding a factor of $\sim 3\text{--}3.5\times$. This gives **$\sim 75\text{--}85$ GPU-hours** per single-step model.
2. **Multi-phase training:** Production models require three training phases: (i) single-step pre-training, (ii) short-horizon autoregressive fine-tuning (4 steps), and (iii) long-horizon autoregressive fine-tuning (up to 12 steps with gradient checkpointing). Phases 2 and 3 add approximately $2\text{--}3\times$ the cost of Phase 1 due to sequential forward passes through the rollout and the memory overhead of gradient checkpointing. Total per model: **~ 250 GPU-hours**.
3. **Three production models:** Precipitation, cloud cover, and solar radiation each require independent training with different input/output configurations. Total: $3 \times 250 =$ **750 GPU-hours**.
4. **Architectural exploration and hyperparameter search:** Each model requires 5–8 exploratory runs to determine optimal architecture depth, attention configuration, loss weighting, learning rate schedule, and data augmentation strategy. This multiplies the per-model cost by $\sim 3\text{--}4\times$. Additional cost: **$\sim 2,000\text{--}2,500$ GPU-hours**.
5. **Ablation studies:** Systematic evaluation of input source contributions (e.g., satellite-only vs. radar+satellite, impact of station observations) requires partial retraining. Estimated: **$\sim 1,500$ GPU-hours**.
6. **Global domain preliminary experiments:** Exploratory training at coarser resolution (8–4 km) on global grids to validate architecture scalability before future full-resolution campaigns. A global grid at 4 km has $\sim 8\text{--}10\times$ more points than CONUS, yielding individual runs of $\sim 500\text{--}800$ GPU-hours. Estimated: **$\sim 2,000\text{--}2,500$ GPU-hours**.
7. **Alternative architectures:** In addition to our baseline U-Net, we plan to evaluate architectures with higher computational cost, such as deeper or wider model variants, non-hierarchical processor-based architectures operating on the full spatial grid, graph neural network approaches, and generative models (e.g., diffusion-based probabilistic nowcasting). These architectures typically involve larger parameter counts ($>100\text{M}$) and higher per-step compute. Estimated: **$\sim 2,500$ GPU-hours**.

8. **Ensemble construction:** Weather forecasting is inherently chaotic: small perturbations in initial conditions can lead to vastly different outcomes, particularly for convective-scale phenomena such as precipitation. To account for this, we plan to train **ensemble forecast models** for each target variable, following best practices in operational numerical weather prediction (e.g., ECMWF ENS, GFS GEFS) and recent ML-based ensemble approaches such as GenCast (Price et al., 2025). Each ensemble member has the same cost as a production model (~ 250 GPU-hours); training 2–3 additional members per variable yields an estimated: **$\sim 2,000$ GPU-hours**.
9. **Data pipeline and evaluation:** Dataset construction, normalization statistics computation, inference benchmarking, and metric evaluation. Estimated: **~ 500 GPU-hours**.

4.6.3 Budget Summary

Activity	GPU-hours (H100)
Production training: 3 continental models at 2 km (3 phases each)	750
Hyperparameter search & architectural exploration	2,250
Ablation studies (input source sensitivity)	1,500
Global domain preliminary experiments (coarser resolution)	2,500
Alternative architectures (non-hierarchical, diffusion models)	2,500
Ensemble construction (3–5 members per variable)	2,000
Data pipeline, statistics & evaluation	500
Contingency for failed runs and debugging ($\sim 20\%$)	3,000
Total	$\sim 15,000$

Table 3: Compute budget breakdown targeting 15,000 GPU-hours on H100.

Each individual training run will typically use 4–8 H100 GPUs with Distributed Data Parallelism (DDP), while smaller experimentation and hyperparameter search jobs will use 1–4 GPUs. We do not plan on using more advanced distributed techniques as our models will fit in the memory provided by a single H100 GPU device. Training uses Automatic Mixed Precision (AMP) with bf16. Autoregressive fine-tuning phases require gradient checkpointing to backpropagate through rollouts of up to 12 steps within H100 device memory.

4.6.4 Resource Consumption Schedule

We plan to consume the allocated resources **uniformly over the allocation period**. The first quarter is dedicated to dataset construction and exploratory experiments, while the remaining three quarters each focus on one of the three target models (precipitation, cloud cover, solar radiation), trained **sequentially and independently**. This sequential approach allows us to incorporate lessons learned from each model into the next and ensures steady, predictable resource utilization throughout the year.

4.7 Licensing, Intellectual Property, and Data Confidentiality

All core training code is proprietary to Odeai. All third-party dependencies (PyTorch, etc.) are open-source with permissive licenses (BSD, MIT, Apache 2.0). Training data sources are publicly available (NOAA GOES, MRMS, HRRR, RTMA, ASOS, ATMS) with no confidentiality constraints. We plan to publish our findings following an **open-science policy**. Trained model weights will be proprietary to Odeai. No GDPR or personal data concerns apply (meteorological observations only).

Quarter	Activity	GPUs	GPU-hours
Q1	Dataset construction at 2 km, pipeline validation, baseline experiments at 4 km, hyperparameter search	1–4	~3,500
Q2	Precipitation model: full training pipeline (3 phases), ablation studies, alternative architecture experiments	4–8	~4,000
Q3	Cloud cover model: full training pipeline, ensemble members, global domain experiments	4–8	~4,000
Q4	Solar radiation model: full training pipeline, ensemble members, final evaluation & benchmarking	4–8	~3,500
Total			~15,000

Table 4: Planned quarterly resource consumption over one year.

References

- R. Lam, A. Sanchez-Gonzalez, M. Willson, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.
- K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619:533–538, 2023.
- J. Pathak, S. Subramanian, P. Harrington, et al. FourCastNet: A global data-driven high-resolution weather forecasting model using adaptive Fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.
- I. Price, A. Sanchez-Gonzalez, F. Alet, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637:84–90, 2025.
- D. Kochkov, J. Yuval, I. Langmore, et al. Neural general circulation models for weather and climate. *Nature*, 632:1060–1066, 2024.
- M. Andrychowicz, L. Espeholt, D. Li, et al. Deep learning for day forecasts from sparse observations. *arXiv preprint arXiv:2306.06079*, 2023.
- X. Zhong, L. Chen, J. Liu, et al. OMG-HD: Observation-driven medium-range global high-resolution weather forecasting. *arXiv preprint arXiv:2407.11710*, 2024.
- M. Assran, Q. Duval, I. Misra, et al. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.
- A. Bardes, Q. Garrido, J. Ponce, X. Chen, M. Rabbat, Y. LeCun, M. Assran, and N. Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.
- R. Balestriero and Y. LeCun. LeJEPa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.