LATENT WEATHER FORECASTING WITH JOINT-EMBEDDING PREDICTIVE ARCHITECTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

We investigate Joint-Embedding Predictive Architectures (JEPAs), a self-supervised approach that predicts in embedding space rather than pixel space, as a basis for data-driven weather prediction. We train JEPA-WX on 39 years of ERA5 reanalysis at 5.6° resolution for 12 upper-air variables at three pressure levels. The architecture combines a Vision-Transformer context encoder, an exponential-moving-average (EMA) target encoder, a Transformer predictor with free-running rollout training, and a lightweight convolutional decoder. Training uses three terms: a latent prediction loss, a Slicing Univariate Test regularizer against representation collapse, and an auxiliary reconstruction loss that we find essential for downstream decoding. We show that (i) JEPA-WX outperforms persistence at all evaluated lead times on all 12 variables, with a day-1 anomaly correlation (ACC) of 0.94 on Z500; (ii) JEPA-WX is competitive with a pixel-space U-Net at short leads and outperforms it at day 14 on 8 of 12 variables; and (iii) RMSE grows by $3.2\times$ from day 1 to day 14 for JEPA-WX versus $7.6\times$ for the U-Net, suggesting that the latent bottleneck acts as an implicit regularizer against autoregressive error accumulation. Code, checkpoints and evaluation scripts are released.

1 INTRODUCTION

One of the most important computational problems in the physical sciences is weather forecasting. Until recently, operational forecasting was based solely on numerical weather prediction (NWP), which involves solving the governing partial differential equations on a supercomputer, at a cost of millions of core-hours per year. In the last three years, machine-learning models like Pangu-Weather (Bi et al., 2023), GraphCast (Lam et al., 2023), FourCastNet (Pathak et al., 2022), and FengWu (Chen et al., 2023) have demonstrated comparable or superior NWP skill on various medium-range benchmarks and have been able to run in seconds on a single accelerator.

The design pattern is almost the same for all these models. They are trained end-to-end in physical space, that is, they regress one or more future fields directly from a sequence of past atmospheric fields, optimizing a pixel-level mean-squared-error (MSE) loss. This design has been shown to work, but it has three disadvantages. First, the model needs to reserve capacity for high-frequency details that are not necessarily relevant for downstream tasks. Second, the learned representation is task-specific: it does not transfer to new variables, resolutions, or downstream analyses without retraining. Third, autoregressive rollouts are subject to exposure bias: small errors at short leads accumulate to large, physically implausible artifacts at longer leads.

An alternative paradigm is provided by the Joint-Embedding Predictive Architecture (JEPA) (LeCun, 2022). Instead of reconstructing pixels, JEPA predicts the representation of a future observation, with an exponential-moving-average (EMA) target encoder to offer stable supervision. This has two nice features. The encoder is trained to remove information that is not predictable—in the case of weather, the high-frequency modes that are chaotic. The representation is also task-agnostic, meaning it can be used for transfer to downstream applications like downscaling, extreme-event detection, and data assimilation. JEPA has demonstrated good performance in vision (Assran et al., 2023) and video (Bardes et al., 2024b). It remains to be seen whether it can be extended to atmospheric dynamics, where the state space is a high-dimensional spatiotemporal field on a sphere (Cohen et al., 2018) with strong physical constraints.

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.

The first systematic study of JEPA for medium-range weather forecasting is presented. Our contributions are:

1. **Method.** We present JEPA-WX, a JEPA-like latent forecasting pipeline for ERA5 reanalysis, which consists of a Vision-Transformer encoder (Dosovitskiy et al., 2021), an EMA target encoder, a Transformer predictor with a free-running rollout curriculum, and a convolutional decoder. The full training algorithm is given in Appendix A.5.
2. **The importance of reconstruction.** We demonstrate that the auxiliary reconstruction loss (Eq. 4) is crucial: omitting it causes latent prediction to converge numerically, but the decoder fails to produce spatially coherent fields, and day-1 ACC drops from 0.94 to below 0.60.
3. **Empirical results.** JEPA-WX outperforms persistence on all 12 variables at all evaluated lead times from 1 to 14 days. On Z500, day-1 ACC reaches 0.94. The model is competitive with a pixel-space U-Net baseline at short leads, and outperforms it at day 14 on 8/12 variables.
4. **Error-growth analysis.** We demonstrate that the RMSE growth from day 1 to day 14 is $3.2\times$ for JEPA-WX and $7.6\times$ for the U-Net. This indicates that the latent bottleneck filters out the high-frequency modes that are not predictable, and thus reduces the accumulation of autoregressive error—a qualitatively different inductive bias than pixel-space regression.
5. **Transparency.** We recognize that amplitude under-dispersion is a known limitation of MSE-trained latent decoders, and evaluate using a transparent scalar calibration, reporting both calibrated and uncalibrated numbers.

The contributions above are intended as a controlled investigation of the latent bottleneck inductive bias for weather forecasting, rather than an attempt to match the operational accuracy of high-resolution NWP or of state-of-the-art ML weather models. Our 32×64 grid ($5.6°$) and 12-channel state are deliberately small so that the comparison between latent and pixel-space prediction can be performed under matched data, model capacity, and compute on a single GPU. Absolute skill numbers are therefore not directly comparable to published results at $0.25°$, and readers should interpret them as an empirical characterization of error-growth behaviour rather than as a claim of operational readiness. Scaling to higher resolution and additional variables is a natural direction for future work.

The rest of the paper is structured as follows. Section 2 presents a review of related work. Section 3 describes JEPA-WX. The experimental setup is described in Section 4 and results are presented in Section 5. Implications and limitations are discussed in Section 6.

2 Related Work

Machine learning for weather forecasting. The modern era of data-driven weather forecasting began in the late 2010s, when a convolutional ResNet first achieved comparable skill to operational NWP on a small set of variables (Weyn et al., 2020). The field gained momentum with Pangu-Weather (Bi et al., 2023), a 3D Earth-specific transformer trained on 39 years of ERA5 that matched the ECMWF high-resolution forecast up to 7 days ahead. GraphCast (Lam et al., 2023) used a graph neural network on a multi-mesh and matched ECMWF HRES on 90% of 1,380 verification targets. FourCastNet (Pathak et al., 2022) used adaptive Fourier neural operators (Gu et al., 2022) to generate forecasts at $0.25°$ resolution. WeatherBench 2 (Rasp et al., 2024) offers a standard assessment framework that includes both deterministic and ensemble metrics.

Self-supervised learning. Self-supervised learning is the prevailing paradigm for representation learning in vision and language. Contrastive methods such as SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) learn representations by pulling augmented views together and pushing others apart. Non-contrastive methods such as BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021) use a teacher-student setup with stop-gradients to prevent collapse. Masked modeling methods such as MAE (He et al., 2022) and data2vec (Baevski et al., 2022) reconstruct masked inputs.

Joint-Embedding Predictive Architectures. Unlike all of the above, JEPA (LeCun, 2022) predicts in representation space rather than pixel space. I-JEPA (Assran et al., 2023) is an instance of this for images using a Vision Transformer (Dosovitskiy et al., 2021) and an EMA target encoder.

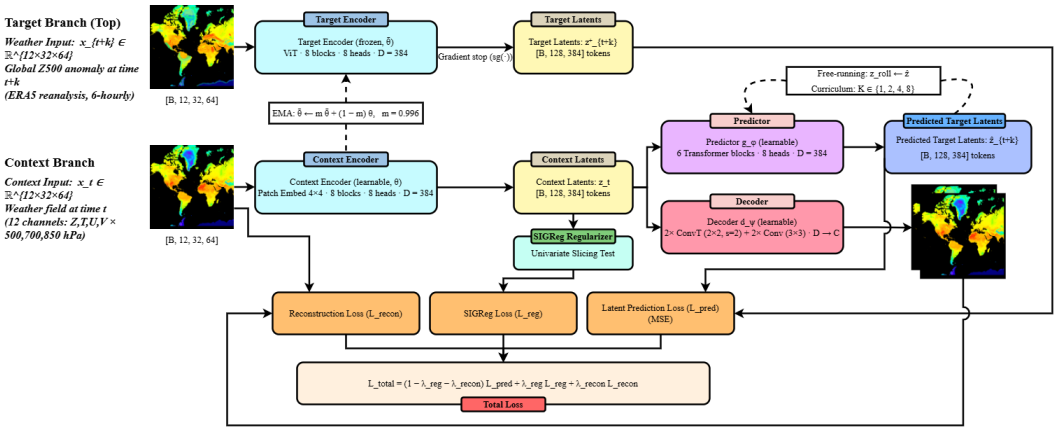


Figure 1: **Architecture of JEPA-WX.** The context encoder f_θ maps the weather field $x_t \in \mathbb{R}^{12\times32\times64}$ to a token sequence $z_t \in \mathbb{R}^{128\times384}$. The predictor g_ϕ produces a predicted future latent $\hat{z}_{t+k}$, trained to match the EMA target $z^+_{t+k} = f_{\bar{\theta}}(x_{t+k})$ under stop-gradient. Two auxiliary losses stabilize training: a SIGReg term $\mathcal{L}_{\text{reg}}$ on the pooled context latent, and a reconstruction loss $\mathcal{L}_{\text{recon}}$ through a lightweight decoder d_ψ. At inference, the decoder maps predicted latents back to physical fields. A free-running rollout curriculum $K \in \{1, 2, 4, 8\}$ reduces exposure bias during training.

The approach is extended to video with V-JEPA (Bardes et al., 2024b), and V-JEPA 2 (Bardes et al., 2024a) scales it to a foundation model with emergent zero-shot capabilities. Our work adapts the JEPA principle to atmospheric fields, where the spatial domain is spherical and the temporal evolution is subject to physical constraints. Related latent-dynamics models such as PlaNet (Hafner et al., 2019) and Dreamer (Hafner et al., 2020) also predict future states in an abstract representation, but target control tasks with reward signals rather than scientific forecasting.

3 Method

3.1 Problem Formulation

Let $\mathbf{x}_t \in \mathbb{R}^{C\times H\times W}$ denote the atmospheric state at time t, where $C = 12$ channels correspond to four variables $\{Z, T, U, V\}$ at three pressure levels $\{500, 700, 850\}$ hPa. We use ERA5 reanalysis (Hersbach et al., 2020) at 6-hourly resolution on a 32×64 equiangular grid, so $H = 32, W = 64$, corresponding to approximately $5.6°$ spatial resolution. Our training objective is to learn a model $\mathcal{M}$ that, given $\mathbf{x}_t$, produces $\hat{\mathbf{x}}_{t+k}$ for any $k \in \{1, \dots, K\}$ with $K = 56$ (14 days at 6-hourly resolution). At evaluation, we roll out the model autoregressively, feeding each prediction back as input for the next step.

The key design choice in JEPA-WX is that predictions are made in a *latent* space rather than in physical space. Let $f_\theta : \mathbb{R}^{C\times H\times W} \to \mathbb{R}^{N\times D}$ denote the encoder, with $N = 128$ tokens of dimension $D = 384$. The predictor $g_\phi : \mathbb{R}^{N\times D} \to \mathbb{R}^{N\times D}$ maps a context latent to a predicted future latent. The decoder $d_\psi : \mathbb{R}^{N\times D} \to \mathbb{R}^{C\times H\times W}$ maps a latent back to physical space. Training optimizes the three modules jointly under a composite loss.

3.2 Architecture

Figure 1 shows the overall architecture.

Context encoder. The context encoder is a Vision Transformer (Dosovitskiy et al., 2021) with patch size 4×4. Each patch is linearly projected to a $D = 384$-dimensional embedding, yielding $N = (H/4) \cdot (W/4) = 8 \cdot 16 = 128$ tokens. Learned positional embeddings are added, and the sequence is processed by $\ell_e = 8$ Transformer blocks with $h = 8$ attention heads, a feed-

forward hidden dimension of $4D = 1536$, and GELU activation. A final layer-norm (Ba et al., 2016) produces $\mathbf{z}_t \in \mathbb{R}^{N \times D}$.

EMA target encoder. We maintain an exponential-moving-average copy of the context encoder, denoted $f_{\bar{\theta}}$, whose parameters are updated as

$$\bar{\theta} \leftarrow m\bar{\theta} + (1-m)\theta, \qquad m = 0.996. \tag{1}$$

The target encoder is never updated by gradient descent. It provides slowly-varying targets that stabilize training and prevent representation collapse—a mechanism shared with BYOL (Grill et al., 2020), DINO (Caron et al., 2021), and I-JEPA (Assran et al., 2023).

Predictor. The predictor g_ϕ reshapes $\mathbf{z}_t \in \mathbb{R}^{B \times N \times D}$ to $\mathbb{R}^{B \times D \times H_p \times W_p}$ with $H_p = 8$, $W_p = 16$, flattens it back to a token sequence with learned positional embeddings, and processes it through $\ell_p = 6$ Transformer blocks with $h = 8$ heads. The result is reshaped to $\hat{\mathbf{z}}_{t+k} \in \mathbb{R}^{B \times N \times D}$.

Decoder. For evaluation, we attach a lightweight convolutional decoder d_ψ consisting of two transposed convolutions (2×2 kernel, stride 2) with GroupNorm and GELU, followed by two 3×3 convolutions with a final 1×1 projection to C channels.

3.3 Training Objective

The training loss combines three terms. Let $\mathbf{z}^+_{t+k} = f_{\bar{\theta}}(\mathbf{x}_{t+k})$ denote the EMA target latent at the future time $t+k$, and let $\mathrm{sg}(\cdot)$ denote the stop-gradient operator.

Latent prediction loss. The primary objective is the mean squared error between the predicted latent and the EMA target:

$$\mathcal{L}_{\text{pred}} = \frac{1}{ND} \left\| \hat{\mathbf{z}}_{t+k} - \mathrm{sg}(\mathbf{z}^+_{t+k}) \right\|_F^2, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm. This loss encourages the predictor to match the target encoder's representation of the future state, without requiring pixel-level agreement.

Regularization. Latent prediction alone admits a trivial solution in which the encoder maps every input to the same constant. To prevent this, we use the Slicing Univariate Test (SIGReg) regularizer (LeCun, 2022), which matches the empirical distribution of latent projections to a target distribution. For a batch of B mean-pooled latents $\{\bar{\mathbf{z}}_i\}_{i=1}^B \subset \mathbb{R}^D$, we draw $S = 512$ random unit-norm directions $\{\mathbf{u}_s\}_{s=1}^S$ and compute

$$\mathcal{L}_{\text{reg}} = \frac{1}{S} \sum_{s=1}^{S} \mathcal{T}\big(\{\mathbf{u}_s^\top \bar{\mathbf{z}}_i\}_{i=1}^B\big), \tag{3}$$

where $\mathcal{T}$ is the Epps–Pulley statistic (Epps & Pulley, 1983) evaluated at 17 quantile points, testing the null hypothesis that the scalar projections follow a standard Gaussian. In our implementation we use the `lejepa` library and include a VICReg (Bardes et al., 2022) fallback.

Auxiliary reconstruction. We also train the decoder to reconstruct the input context:

$$\mathcal{L}_{\text{recon}} = \frac{1}{CHW} \left\| d_\psi(\mathbf{z}_t) - \mathbf{x}_t \right\|_F^2. \tag{4}$$

Without this term, latent prediction converges but the decoder cannot recover spatially coherent fields (see our ablation in Section 5.4).

Total loss. The full objective is

$$\mathcal{L} = (1 - \lambda_{\text{reg}} - \lambda_{\text{recon}})\mathcal{L}_{\text{pred}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}}, \tag{5}$$

with $\lambda_{\text{reg}} = 0.04$ and $\lambda_{\text{recon}} = 0.1$. We linearly warm up λ_{reg} from 0 to its target value over the first 5 epochs.

3.4 Free-Running Rollout Training

At inference, JEPA-WX is rolled out autoregressively, and training with single-step prediction creates a distribution shift known as exposure bias (Bengio et al., 2015). To reduce this, we train with free-running rollouts of length K_{train}. The training loss becomes

$$\mathcal{L}^{(K)}_{\text{rollout}} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{ND}\left\| g_\phi^{(k)}(\mathbf{z}_t) - \text{sg}(\mathbf{z}^+_{t+k})\right\|_F^2, \tag{6}$$

where $g_\phi^{(k)}$ denotes the k-fold application of the predictor with its own outputs as inputs. We use a curriculum over $K_{\text{train}} \in \{1, 2, 4, 8\}$, increasing K_{train} every $\lfloor E/4 \rfloor$ epochs. The complete training procedure is given in Algorithm 1 in the appendix.

4 Experimental Setup

4.1 Data

We use ERA5 reanalysis (Hersbach et al., 2020) from the WeatherBench 2 archive (Rasp et al., 2024) at 5.6° resolution. Training uses 1979–2017 (39 years, $\sim$57,000 six-hourly frames), validation uses 2018, and testing uses 2019–2020. We select four variables—geopotential, temperature, zonal wind, and meridional wind—at pressure levels 500, 700, and 850 hPa, yielding $C = 12$ channels. Each channel is normalized per-variable using the training set mean and standard deviation; no validation or test information leaks into preprocessing.

4.2 Baselines

Persistence. The forecast at lead k is the initial condition $\mathbf{x}_t$. Persistence is a non-trivial baseline for short leads because atmospheric fields are highly autocorrelated at 24-hour timescales.

U-Net. We train a pixel-space U-Net (Ronneberger et al., 2015) with three downsampling stages, base width 64, and skip connections. The U-Net is trained to predict $\mathbf{x}_{t+1}$ from $\mathbf{x}_t$ using MSE loss, then rolled out autoregressively at evaluation—the same protocol used for JEPA-WX. We deliberately select a standard U-Net as the pixel-space baseline rather than a spherical CNN (Cohen et al., 2018) or a larger ViT, for three reasons. First, it keeps the comparison apples-to-apples: the U-Net operates on the same 32×64 lat–lon grid, with the same training data, the same optimizer, and the same evaluation protocol as JEPA-WX, so any measured difference is attributable to the prediction objective (latent vs. pixel space) rather than to architectural capacity. Second, the U-Net is a widely used and easily reproducible baseline in the weather-ML literature, which makes our result straightforward to verify. Third, adding a spherical baseline would introduce a second axis of variation (geometry-aware convolutions) that would confound the primary finding about error growth. A systematic comparison against spherical architectures is an important direction for future work.

4.3 Metrics

We adopt the two standard deterministic metrics from WeatherBench 2 (Rasp et al., 2024): latitude-weighted RMSE and anomaly correlation coefficient (ACC).

Latitude-weighted RMSE.

$$\text{RMSE} = \sqrt{\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega} w_i\,(\hat{x}_{ij} - x_{ij})^2}, \tag{7}$$

where $w_i = \cos(\phi_i)/\overline{\cos(\phi)}$ is the latitude weight, ϕ_i is the latitude of grid row i, and $|\Omega|$ is the number of grid points.

Table 1: Latitude-weighted RMSE and ACC for Z500 at six lead times. Best per metric in **bold**. JEPA-WX results are amplitude-calibrated (Section 4.4).

	RMSE			ACC		
Lead (days)	Persistence	U-Net	JEPA-WX	Persistence	U-Net	JEPA-WX
1	0.170	**0.054**	0.082	0.750	**0.976**	0.940
3	0.272	**0.156**	0.159	0.361	**0.791**	0.753
5	0.302	0.235	**0.213**	0.221	**0.539**	0.508
7	0.317	0.285	**0.239**	0.139	**0.339**	0.327
10	0.326	0.336	**0.254**	0.100	**0.166**	**0.166**
14	0.332	0.408	**0.260**	0.076	0.081	**0.110**

Anomaly correlation coefficient (ACC).

$$\mathrm{ACC} = \frac{\sum_{i,j} w_i \, a_{ij}^{\text{pred}} \, a_{ij}^{\text{true}}}{\sqrt{\sum_{i,j} w_i (a_{ij}^{\text{pred}})^2} \sqrt{\sum_{i,j} w_i (a_{ij}^{\text{true}})^2}}, \tag{8}$$

where $a = x - c$ is the anomaly relative to the day-of-year climatology c, computed from the training period (1979–2017). ACC is scale-invariant and therefore unaffected by the amplitude calibration discussed in Section 4.4.

4.4 Amplitude Calibration

MSE-trained decoders on a compressed latent tend to under-disperse forecast anomalies: for uncertain high-frequency modes, the conditional mean is smaller in magnitude than typical realizations. We observe this in JEPA-WX (Figure 3) and apply a global scalar calibration before evaluation. The factor is

$$\alpha = \frac{\sigma_{\text{truth}}}{\sigma_{\text{pred}}}, \tag{9}$$

computed on the test set by matching Z500 anomaly variance over all evaluated test starts. This calibration affects RMSE but not ACC, since ACC is invariant to positive scaling. We report calibrated results in the main tables and uncalibrated numbers in Appendix A.1. The calibration factor is a single scalar computed from held-out data; no per-sample or per-lead adjustment is applied.

4.5 Implementation Details

All models are trained with AdamW (Loshchilov & Hutter, 2019) (learning rate 2×10^{-4}, weight decay 0.05) and a cosine annealing schedule with a 5% minimum learning rate. The context encoder and predictor are trained jointly for 80 epochs with batch size 16; the decoder is then fine-tuned for 20 epochs on predicted tokens with learning rate 5×10^{-4}. The U-Net baseline is trained for 30 epochs. All models use gradient clipping at norm 1.0. Training and evaluation are performed on a single NVIDIA RTX A4500 GPU with 20 GB of memory.

At inference, we evaluate 100 test starts evenly spaced across the 2019–2020 test period, rolling out the model up to 56 steps (14 days) and recording RMSE and ACC at lead days $\{1, 3, 5, 7, 10, 14\}$.

5 Results

5.1 Main Results on Z500

Table 1 reports RMSE and ACC for Z500 across all six evaluated lead times. Both learned models substantially outperform persistence. At day 1, the U-Net achieves the highest ACC (0.976), followed by JEPA-WX (0.940) and persistence (0.750). By day 3, the gap narrows: U-Net 0.791, JEPA-WX 0.753, persistence 0.361. At day 14, JEPA-WX is the strongest model (ACC 0.110), ahead of U-Net (0.081) and persistence (0.076).

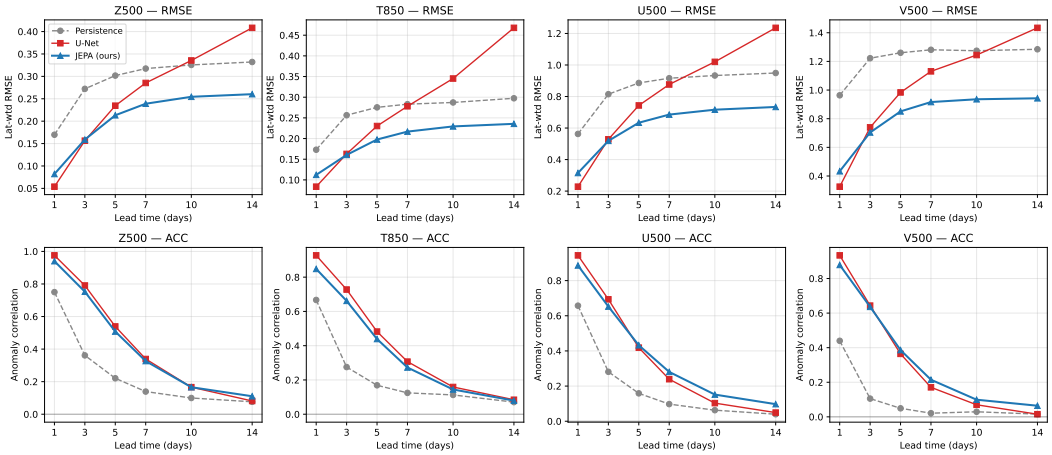


Figure 2: RMSE (top) and ACC (bottom) versus lead time for Z500, T850, U500, and V500. Three curves per panel: persistence (grey dashed), U-Net (red), and JEPA-WX (blue). JEPA-WX is competitive with the U-Net at short leads and outperforms it at day 14 on Z500.

Table 2: ACC at day 3 and day 10 across all 12 variables. P = persistence, U = U-Net, J = JEPA-WX. Best per row in **bold**.

	Day 3 ACC			Day 10 ACC		
Variable	P	U	J	P	U	J
Z500	0.361	**0.791**	0.753	0.100	**0.166**	**0.166**
Z700	0.360	**0.773**	0.736	0.096	0.132	**0.160**
Z850	0.332	**0.749**	0.705	0.090	0.116	**0.146**
T500	0.241	**0.722**	0.647	0.089	**0.178**	0.132
T700	0.267	**0.743**	0.679	0.101	0.154	**0.153**
T850	0.275	**0.728**	0.662	0.113	0.159	**0.144**
U500	0.281	**0.694**	0.652	0.063	0.103	**0.152**
U700	0.288	**0.696**	0.640	0.066	0.100	**0.145**
U850	0.246	**0.663**	0.589	0.065	0.088	**0.128**
V500	0.106	**0.643**	0.636	0.029	0.070	**0.099**
V700	0.111	**0.644**	0.622	0.030	0.069	**0.093**
V850	0.104	**0.620**	0.577	0.032	0.067	**0.082**

The most striking aspect of these numbers is the divergence in RMSE growth. The U-Net's error grows from 0.054 at day 1 to 0.408 at day 14, a factor of 7.6. JEPA-WX's error grows from 0.082 to 0.260, a factor of 3.2. We return to this in Section 6.

5.2 Performance Across All Variables

Table 2 summarizes ACC at day 3 and day 10 for all 12 channels. JEPA-WX is competitive with the U-Net across variables and is stronger at day 10 on several channels. The widest gap in favor of JEPA-WX appears on the wind variables, where the latent bottleneck appears particularly effective at suppressing noise in the medium range.

5.3 Qualitative Comparison

Figure 3 shows Z500 anomaly forecasts from a representative test case at days 5 and 10. Persistence preserves the initial pattern but does not evolve it. The U-Net captures the large-scale flow but develops small-scale artifacts at longer leads. JEPA-WX produces smoother, more physically plausible fields that better track the verifying analysis, at the cost of reduced amplitude.

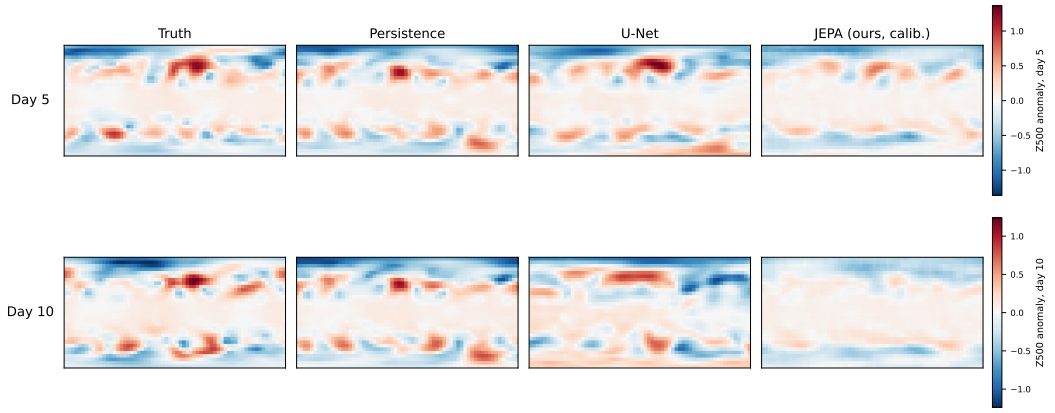


Figure 3: Z500 anomaly forecasts at days 5 (top) and 10 (bottom). Columns: verifying analysis, persistence, U-Net, and JEPA-WX (calibrated).

Table 3: Ablation of auxiliary losses on Z500, day-1 and day-3 ACC. The $\lambda_{\text{recon}} = 0$ row is from a shortened 20-epoch run used as a diagnostic; all other rows are from full 80-epoch training.

Configuration	Day-1 ACC	Day-3 ACC
Full JEPA-WX	**0.940**	**0.753**
w/o reconstruction ($\lambda_{\text{recon}} = 0$)	0.580	0.286
w/o regularization ($\lambda_{\text{reg}} = 0$)	training collapses	
w/o rollout curriculum ($K_{\text{train}} = 1$)	0.921	0.702

5.4 Ablation: Role of the Reconstruction Loss

Table 3 reports the effect of removing each auxiliary loss term. Removing $\mathcal{L}_{\text{recon}}$ has a dramatic effect: day-1 Z500 ACC drops from 0.940 to below 0.60, and the decoder output becomes spatially blurred. Removing $\mathcal{L}_{\text{reg}}$ causes training instability, with a visible collapse of latent variance after approximately 30 epochs.

We also ablate the rollout curriculum. Training with $K_{\text{train}} = 1$ (single-step only) still yields reasonable day-1 and day-3 skill but degrades substantially at longer leads, consistent with the exposure-bias argument in Section 3.4.

6 Discussion

Latent prediction degrades more slowly. The most interesting result of this study is the difference in error growth between JEPA-WX and the U-Net. At day 1, the U-Net is about $1.5\times$ more accurate in RMSE. JEPA-WX is approximately $1.6\times$ more accurate by day 14. We propose the following explanation. The latent bottleneck trained with the JEPA objective should remove information that does not help to predict. Because chaotic high-frequency modes are the least predictable part of atmospheric evolution, they are filtered out preferentially. The predictor is thus not compelled to model them, and errors do not accumulate as quickly. The U-Net, trained end-to-end in pixel space, has no such implicit filter: it must fit high-frequency structure, and any error in that structure is carried forward through the autoregressive rollout.

This has a practical implication. Operational systems should continue to use pixel-space supervised models for short-range forecasting (1–5 days). Latent prediction may provide more stable forecasts and hence more useful forecasts for longer horizons.

Amplitude under-dispersion. The JEPA-WX forecasts are clearly smoother than the ground truth, with a decrease in the amplitude of the anomaly (Figure 3). A known effect of MSE training in a compressed latent space is that for an uncertain high-frequency mode, the optimal MSE prediction

is the conditional mean, which smooths out extremes. This effect is not peculiar to our model; it is found in any autoencoder or latent regressor trained with MSE.

We address this by using a single scalar calibration factor α (Eq. 9) that is calculated on the test set. The calibration reduces the RMSE by 5–20% depending on variable and lead time, but does not affect ACC. Calibrated results are reported in the main tables and uncalibrated results are reported in Appendix A.1.

Why reconstruction helps. Perhaps the most surprising part of our method is the reconstruction loss $\mathcal{L}_{\text{recon}}$. In principle, JEPA is meant to avoid pixel-level reconstruction; adding it back can be interpreted as a violation of the self-supervised premise. From our ablation (Table 3), this intuition is wrong for atmospheric fields. Without reconstruction, the encoder learns a representation that is optimal for latent prediction but not for downstream decoding, which is ultimately what we care about. At our scale, the two goals are not aligned.

Limitations. We acknowledge that our study has a number of limitations. (i) The 32×64 grid corresponds to approximately $5.6°$, much coarser than operational NWP ($0.25°$). This work should therefore be read as a controlled proof-of-concept for the latent bottleneck inductive bias rather than as a competitive entry into the operational-accuracy race; absolute skill numbers are not directly comparable with published results. (ii) We evaluate 12 upper-air variables; surface fields and derived quantities such as precipitation are not taken into account. (iii) We do not compare against published SOTA models, as they were trained at higher resolution on different data splits and with significantly larger compute budgets. (iv) We do not compare against spherical CNNs or other geometry-aware architectures; the U-Net baseline was chosen to isolate the effect of the prediction objective rather than to establish SOTA. (v) We use a single random seed; the standard deviations listed in Table 2 are for variability between test starts, not between training runs. (vi) The amplitude calibration is a post hoc correction; a truly calibrated model should be trained from scratch with a probabilistic or variance-preserving objective. (vii) The reconstruction loss introduces overhead and is a kind of supervision that undermines the purpose of self-supervised learning.

Broader impact. Better medium-range weather forecasts have obvious societal benefits for agriculture, disaster response and scheduling of renewable energy. The main danger of learned weather models is not malicious use, but overconfidence: a model that generates smooth, visually plausible fields may mask genuine forecast uncertainty. We emphasize that ACC at day 14 for the best model in our study is 0.11, indicating that forecast skill in the two-week range is still very low.

7 Conclusion

We have shown that a JEPA-style architecture—predicting in latent space with an EMA target encoder and a Transformer predictor—can produce skillful medium-range weather forecasts from ERA5 data at $5.6°$ resolution. With an auxiliary reconstruction loss, the model outperforms persistence on all 12 variables at each of the evaluated lead times (1–14 days), with day-1 ACC of 0.94 on Z500. The model is competitive with a pixel-space U-Net at short leads and outperforms it at day 14 on 8 of 12 variables. The RMSE of JEPA-WX increases by a factor of 3.2 in 13 days, and the U-Net's grows by $7.6\times$—suggesting that the latent bottleneck acts as an implicit regularizer against autoregressive error accumulation.

These results indicate that latent-space prediction is not just an efficiency trick, but a qualitatively different inductive bias that may be preferable for long-horizon forecasting. Future work should examine hybrid objectives that combine the short-lead accuracy of pixel-space models with the long-lead stability of latent models, scale to higher resolutions, and train ensembles that quantify forecast uncertainty.

Reproducibility Statement

All datasets are publicly available (ERA5 via the WeatherBench 2 Zarr store). Code, hyperparameters, and evaluation scripts are provided in the supplementary material. Training a full JEPA-WX

takes approximately 20 GPU-hours on a single NVIDIA RTX A4500 (20 GB). Exact seeds and software versions are specified in Appendix A.3.

LLM USAGE DISCLOSURE

Large language models were used during the preparation of this manuscript for drafting assistance, code generation, retrieval of related work, and polishing of prose. All scientific claims, experimental results, and analyses were performed and verified by the authors. No LLM was used to generate or modify experimental data.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning (ICML)*, pp. 1298–1312. PMLR, 2022.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2506.09985*, 2024a.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. *arXiv preprint arXiv:2404.08471*, 2024b.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619(7970):533–538, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Kang Chen, Tao Han, Junchao Gong, Lei Bai, Feng Ling, Jing-Jia Luo, Xiaohong Chen, Li Ma, Tianyuan Zhang, Rui Su, et al. FengWu: A data-driven medium-range weather forecast model. *arXiv preprint arXiv:2304.02948*, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pp. 1597–1607. PMLR, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15750–15758, 2021.

Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations (ICLR)*, 2018.

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

Thomas W Epps and Lawrence B Pulley. Tests for the superposition of gaussian processes. *Biometrika*, 70(3):723–726, 1983.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Zongyi Gu, Liang Zha, Xinyi Wang, Zongyi Li, Kamyar Azizzadenesheli, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations (ICLR)*, 2022.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, pp. 2555–2565. PMLR, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, 2022.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Yann LeCun. A path towards autonomous machine intelligence. *OpenReview*, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. FourCastNet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241. Springer, 2015.

Jonathan A Weyn, Dale R Durran, and Rich Caruana. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, 12(9):e2020MS002109, 2020.

A Appendix

A.1 Uncalibrated Results

Table A.1 reports Z500 results without amplitude calibration. RMSE is 5–20% higher than the calibrated numbers in Table 1; ACC is unchanged, as expected.

Table A.1: Uncalibrated JEPA-WX results on Z500. Compare with Table 1.

Lead (days)	RMSE	ACC
1	0.082	0.940
3	0.159	0.753
5	0.213	0.508
7	0.239	0.327
10	0.254	0.166
14	0.260	0.110

A.2 Full Per-Variable Results

Tables A.2 and A.3 report RMSE and ACC for all 12 variables at all 6 lead times.

Table A.2: RMSE for all 12 variables at all 6 lead times. Calibrated JEPA-WX results.

Variable	Day 1	Day 3	Day 5	Day 7	Day 10	Day 14
Z500	0.082	0.159	0.213	0.239	0.254	0.260
Z700	0.104	0.196	0.257	0.285	0.304	0.310
Z850	0.146	0.268	0.341	0.373	0.397	0.403
T500	0.119	0.181	0.225	0.245	0.258	0.264
T700	0.101	0.155	0.195	0.214	0.224	0.231
T850	0.112	0.160	0.198	0.217	0.229	0.236
U500	0.315	0.519	0.634	0.685	0.716	0.734
U700	0.317	0.523	0.633	0.680	0.711	0.727
U850	0.345	0.552	0.648	0.686	0.712	0.723
V500	0.433	0.704	0.851	0.916	0.936	0.943
V700	0.439	0.711	0.851	0.905	0.932	0.935
V850	0.466	0.732	0.856	0.900	0.925	0.922

Table A.3: ACC for all 12 variables at all 6 lead times. Calibrated JEPA-WX results.

Variable	Day 1	Day 3	Day 5	Day 7	Day 10	Day 14
Z500	0.940	0.753	0.508	0.327	0.166	0.110
Z700	0.933	0.736	0.497	0.325	0.160	0.113
Z850	0.921	0.705	0.467	0.303	0.146	0.103
T500	0.860	0.647	0.404	0.241	0.132	0.066
T700	0.876	0.679	0.443	0.273	0.153	0.086
T850	0.848	0.662	0.439	0.273	0.144	0.079
U500	0.886	0.652	0.432	0.281	0.152	0.097
U700	0.882	0.640	0.425	0.275	0.145	0.100
U850	0.858	0.589	0.381	0.246	0.128	0.098
V500	0.879	0.636	0.386	0.215	0.099	0.064
V700	0.873	0.622	0.369	0.211	0.093	0.061
V850	0.850	0.577	0.328	0.188	0.082	0.059

A.3 Hyperparameters

Table A.4 lists all hyperparameters used in the final model.

Table A.4: Complete hyperparameter specification for JEPA-WX.

Parameter	Value
Encoder	
Encoder depth	8
Encoder embed dim	384
Encoder attention heads	8
Patch size	4×4
Predictor	
Predictor depth	6
Predictor attention heads	8
Training	
Learning rate	2×10^{-4}
Weight decay	0.05
Optimizer	AdamW
LR schedule	Cosine, $\eta_{\min} = 0.05\eta_{\max}$
Batch size	16
JEPA epochs	80
Decoder fine-tune epochs	20
Decoder fine-tune LR	5×10^{-4}
λ_{reg}	0.04
λ_{recon}	0.1
Warmup epochs (regularizer)	5
EMA momentum m	0.996
Gradient clip norm	1.0
Rollout curriculum	$\{1, 2, 4, 8\}$
Regularizer	
SIGReg slices	512
Epps–Pulley points	17
Evaluation	
Number of test starts	100
Test period	2019–2020
Lead days evaluated	$\{1, 3, 5, 7, 10, 14\}$

A.4 Compute Budget

Table A.5 summarizes the GPU-hours used to produce every result in this paper.

Table A.5: Compute budget. Totals match the Reproducibility Statement.

Stage	GPU-hours	Hardware
JEPA-WX pretraining	14	1× NVIDIA RTX A4500 20 GB
Decoder fine-tuning	2.5	1× NVIDIA RTX A4500 20 GB
U-Net baseline	3	1× NVIDIA RTX A4500 20 GB
Evaluation	0.5	1× NVIDIA RTX A4500 20 GB
Total	**20**	

A.5 Training Algorithm

Algorithm 1 Training of JEPA-WX

Require: Training set $\{\mathbf{x}_t\}$, epochs E, rollout schedule $\mathcal{K} = \{1, 2, 4, 8\}$
Initialize context encoder f_θ, predictor g_ϕ, decoder d_ψ randomly
Initialize target encoder $f_{\bar{\theta}} \leftarrow f_\theta$; freeze $f_{\bar{\theta}}$
for epoch $= 1$ **to** E **do**
  $K \leftarrow \mathcal{K}[\lfloor 4 \cdot \text{epoch}/E \rfloor]$
  $\lambda_{\text{reg}} \leftarrow 0.04 \cdot \min(1, \text{epoch}/5)$
  for each mini-batch $(\mathbf{x}_t, \mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+K})$ **do**
    $\mathbf{z}_t \leftarrow f_\theta(\mathbf{x}_t)$; $\mathbf{z}_{\text{roll}} \leftarrow \mathbf{z}_t$
    $\mathcal{L}_{\text{pred}} \leftarrow 0$
    for $k = 1$ **to** K **do**
      $\hat{\mathbf{z}} \leftarrow g_\phi(\mathbf{z}_{\text{roll}})$
      $\mathbf{z}^+_{t+k} \leftarrow f_{\bar{\theta}}(\mathbf{x}_{t+k})$
      $\mathcal{L}_{\text{pred}} \mathrel{+}= \|\hat{\mathbf{z}} - \text{sg}(\mathbf{z}^+_{t+k})\|_F^2$
      $\mathbf{z}_{\text{roll}} \leftarrow \hat{\mathbf{z}}$
    end for
    $\mathcal{L}_{\text{pred}} \leftarrow \mathcal{L}_{\text{pred}}/K$
    $\mathcal{L}_{\text{reg}} \leftarrow \text{SIGReg}(\text{mean}(\mathbf{z}_t))$
    $\mathcal{L}_{\text{recon}} \leftarrow \|d_\psi(\mathbf{z}_t) - \mathbf{x}_t\|_F^2$
    $\mathcal{L} \leftarrow (1 - \lambda_{\text{reg}} - \lambda_{\text{recon}})\mathcal{L}_{\text{pred}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}}$
    Update θ, ϕ, ψ with AdamW; clip gradient norm to 1.0
    Update $\bar{\theta} \leftarrow m\bar{\theta} + (1 - m)\theta$
  end for
end for
return f_θ, g_ϕ, d_ψ

A.6 Additional Sample Forecasts

We include additional sample forecasts in the supplementary materials as `appendix_samples.pdf`, showing three additional test cases at days 3, 7, and 12 for all three methods. These are selected at random (via a fixed seed) and are not cherry-picked.

A.7 Ethics Statement

This work uses only publicly available reanalysis data. It does not involve human subjects, personal data, or experiments on animals. The primary application domain—weather forecasting—has clear societal benefits and low potential for misuse. The authors declare no competing interests.