SPANWORLD: A VARIABLE-SPAN WORLD MODEL FOR VEGETATION FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

World models learn internal representations of environments and their evolution, providing a way to forecast future environmental change. These models need to preserve the predictive value of their internal representations as rollouts progress. Evaluating this capability requires examining continued predictions from generated states alongside direct forecasts from history. To this end, we propose SpanWorld, a remote-sensing world model that learns vegetation dynamics from satellite observations and advances its latent spatial state through weather-conditioned transitions. The model uses given future weather as an external driver, and its subsequent predictions are evaluated against real remote-sensing imagery. SpanWorld combines baseline predictions derived from incomplete historical observations with a dynamic spatial state updated by a shared variable-span transition, supporting direct prediction and partitioned rollout within a finite forecast window. We further introduce a partitioned-path learning mechanism that retains supervision of predictions made directly from the same initial state at every future target date. It also samples temporal partitions and uses observed target images to supervise predictions made by continuing from model-generated intermediate states. On GreenEarthNet's 100-day dense normalized difference vegetation index (NDVI) image forecasting task, we compare SpanWorld with a control using the same architecture. Both models receive supervision of direct predictions at all forecast horizons. Using the same endpoint images, we additionally supervise predictions from partitioned rollouts in SpanWorld and direct predictions in the control. On the IID split, both models roll out to day 50 using one temporal partition unseen during training, then predict the image at each target date from day 55 to day 100 separately from their respective day-50 states. For these subsequent predictions, partitioned-path learning reduces mean squared error (MSE) by approximately 11% relative to the control, while direct-prediction errors over the same period remain similar.

1 INTRODUCTION

World models provide a foundation for prediction, planning, and decision-making by simulating environmental dynamics. These simulations help anticipate how an environment may evolve and inform subsequent actions. By learning internal representations and their dynamics from observations, world models enable possible future changes to be explored within the model.

Existing approaches use learned latent dynamics models to generate future trajectories for behavior learning and planning (Hafner et al., 2020; Hansen et al., 2024). In models that perform continued rollout through state transitions, generated states serve as inputs to subsequent transitions to predict further into the future (Hafner et al., 2021). This distinguishes prediction at a target time from continued prediction using the state reached at that time. The former assesses the model's output at the current target, while the latter examines whether the generated state retains information useful for subsequent prediction. It is therefore worth examining how training constrains the use of generated states and how these constraints affect subsequent prediction.

Weather-driven vegetation change provides a concrete setting for examining the predictive role of generated states. Previous studies combine remote-sensing observations, meteorological inputs, and spatial information to forecast vegetation change, with more recent work using weather-conditioned

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.

recurrent latent dynamics to simulate its evolution (Benson et al., 2024; Iele et al., 2026). This setting brings together complementary observations and external drivers. Remote-sensing imagery records the spatial distribution of vegetation and its changes over time, but historical observations are often incomplete, requiring predictive representations to be formed from limited observations. Time-varying weather provides an external driver of subsequent evolution, while later imagery provides observations against which predictions can be evaluated. With future weather given and no additional remote-sensing imagery supplied during prediction, these conditions allow the predictive value of model-generated states to be examined against observed vegetation change.

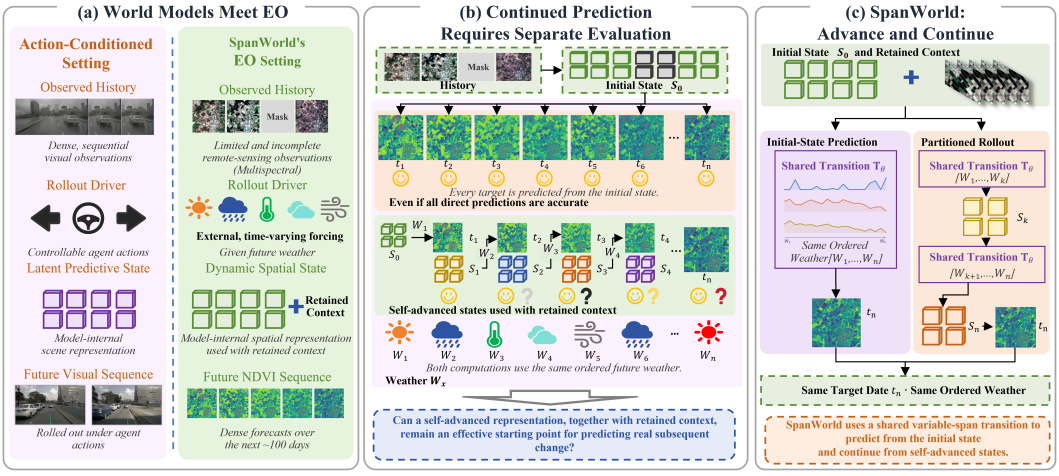


Figure 1: **World-model setting and the continuation question.** (a) Action-conditioned visual prediction and weather-conditioned vegetation forecasting. (b) Even with accurate direct predictions, continued prediction from generated states requires separate evaluation. (c) SpanWorld uses shared dynamics, ordered weather, and retained historical context for direct and continued prediction without new remote-sensing imagery.

With the shared variable-span transition considered here, a model can continue forecasting over intervals of different lengths. The same target date can therefore be reached either directly from the initial state or through intermediate states along a partitioned rollout. Our central question in this setting is: **When training already supervises direct predictions from the same initial state at every future target date, is this sufficient for states generated through partitioned rollout to support accurate subsequent prediction?** This supervision is applied to predictions made from the initial state. We test whether this supervision is sufficient for continued prediction by using model-generated states to predict later observations. The dates covered by supervision and the state-transition paths along which it is applied therefore need to be examined separately (Figure 1).

To address this question, we introduce SpanWorld, a remote-sensing world model for vegetation forecasting. SpanWorld uses a shared, weather-conditioned variable-span transition to support direct prediction of vegetation imagery and partitioned rollout within a finite forecast window. We further introduce a partitioned-path learning mechanism that uses subsequent prediction quality to guide state learning. This mechanism retains supervision of direct predictions and uses observed endpoint imagery to supervise predictions made by continuing from model-generated states. Controlled comparisons using the same architecture and endpoint images for supervision show that this mechanism reduces degradation in continued prediction after rollout while direct-prediction errors remain similar.

Our main contributions are as follows:

1. **We introduce SpanWorld, a remote-sensing world model for vegetation forecasting.** Its shared weather-conditioned variable-span transition updates a dynamic spatial state, supporting direct prediction of dense vegetation imagery and partitioned rollout within a finite forecast window.
2. **We introduce a learning mechanism that uses real observations to supervise predictions from partitioned rollout.** The mechanism retains supervision of direct predictions

while using observed endpoint imagery to supervise predictions produced through different temporal partitions. This incorporates the quality of subsequent predictions from generated intermediate states into training.

3. **We reveal how the choice of supervision path affects continued prediction in SpanWorld.** The controlled comparison holds the architecture and supervision of direct predictions at all forecast horizons fixed. Both schemes use the same endpoint images for additional supervision, applied to either direct or partitioned predictions. Under the evaluated conditions, partitioned-path supervision improves SpanWorld's continued prediction from generated states, while direct-prediction errors remain similar between the two training schemes.

2 Related Work

World models learn representations and dynamics of environments from existing observations, providing a foundation for future prediction, planning, and behavior learning. In model-based control and policy learning, Dyna, PETS, and MBPO respectively demonstrate the roles of model-assisted planning, probabilistic dynamics, and model-generated trajectories (Sutton, 1991; Chua et al., 2018; Janner et al., 2019). In visual environments, E2C, World Models, SimPLe, PlaNet, and the Dreamer series support control, planning, and behavior learning through pixel or latent-state prediction (Watter et al., 2015; Ha & Schmidhuber, 2018; Kaiser et al., 2020; Hafner et al., 2019; 2020; 2021; 2023), while DeepMDP, MuZero, and the TD-MPC series organize state representation and dynamics learning around task-relevant information (Gelada et al., 2019; Schrittwieser et al., 2020; Hansen et al., 2022; 2024). For sequence modeling, IRIS, TWM, and STORM use Transformers to model temporal dynamics, DIAMOND introduces diffusion-based generation, and TWISTER adopts multi-step contrastive prediction (Micheli et al., 2023; Robine et al., 2023; Zhang et al., 2023; Alonso et al., 2024; Burchi & Timofte, 2025). Visual representation learning has also expanded toward reconstruction-free objectives and dynamics prediction over pretrained features: DreamerPro uses prototypical representation learning, whereas DINO-WM and V-JEPA 2-AC learn dynamics over pretrained visual representations and support planning (Deng et al., 2022; Zhou et al., 2025; Assran et al., 2025). Across these approaches, internal representations support prediction or decision making while carrying information needed for further rollout. Prediction accuracy from observations and the ability of model-generated representations to support subsequent prediction are therefore related but distinct questions.

Weather-conditioned Earth observation forecasting connects learned dynamics with the evolution of real environments. Methods based on recurrent memory, structured decomposition, convolutional prediction, conditional diffusion, and spatiotemporal attention have provided methodological foundations for video and Earth-system forecasting (Shi et al., 2015; Wang et al., 2017; 2018; Le Guen & Thome, 2020; Gao et al., 2022a;b; Voleti et al., 2022). EarthNet2021 established satellite-image forecasting guided by future weather, and subsequent work further examined the predictive role of weather information and its value for conditional simulation (Requena-Mesa et al., 2021; Diaconu et al., 2022). GreenEarthNet and Contextformer combine historical imagery, meteorology, and spatial conditions for dense NDVI forecasting (Benson et al., 2024); VegeDiff models uncertainty in vegetation change with latent diffusion to generate future vegetation imagery (Zhao et al., 2025); and VegSim further develops a vegetation world model conditioned on weather scenarios, using recurrent latent dynamics to generate probabilistic NDVI trajectories (Iele et al., 2026). Together, these advances extend the scope of Earth observation forecasting from future-image estimation to weather-conditioned dynamics simulation. Incomplete observation histories, time-varying weather, and spatial heterogeneity make vegetation forecasting a setting that calls for both representation learning and conditional dynamics modeling. Building on this foundation, we examine whether representations formed through a model's own rollout can support subsequent vegetation forecasting under given future weather, without additional remote-sensing observations.

Multi-step prediction and dynamics learning study how training objectives and rollout processes can preserve information useful for the future. Scheduled Sampling addresses the mismatch between training and generation by sampling between observed and model-generated inputs, while Professor Forcing aligns recurrent dynamics under teacher forcing and free running (Bengio et al., 2015; Lamb et al., 2016). CPC, SPR, and TWISTER construct learning objectives around predicting future representations, while latent overshooting in PlaNet and representation-consistency

constraints in EfficientZero further provide learning signals for multi-step rollout (van den Oord et al., 2018; Schwarzer et al., 2021; Burchi & Timofte, 2025; Hafner et al., 2019; Ye et al., 2021). In learned physical simulation and neural PDE solving, perturbing states during training, training on model-generated states, and refining predictions have also been used to improve rollout stability and prediction quality (Sanchez-Gonzalez et al., 2020; Brandstetter et al., 2022; Lippe et al., 2023). For temporal and evolution-operator modeling, Neural ODE and Latent ODE describe continuous-time state evolution, while DeepONet and the Fourier Neural Operator learn mappings in function spaces (Chen et al., 2018; Rubanova et al., 2019; Lu et al., 2021; Li et al., 2021). Deep-OSG further learns variable-time-step evolution operators for autonomous dynamical systems, studies semigroup consistency between direct and composed evolution, and includes a training formulation that uses observed endpoint states to supervise both direct and composed predictions (Chen & Wu, 2023). Building on these foundations, SpanWorld combines a weather-conditioned shared variable-span state transition with predictive learning on model-generated states. While retaining supervision of direct predictions at all forecast horizons, partitioned-path learning additionally supervises predictions from model-generated states using observed imagery at the target dates. This trains the same model both to predict directly from historical observations and to continue forecasting from intermediate representations formed through its own rollout.

3 METHOD

SpanWorld is designed to support accurate subsequent prediction from states generated during its own rollout. Direct supervision across the forecast window trains the model to predict each future date from the initial state. We jointly design the state transition and its supervision to support continued prediction from generated states. A shared, weather-conditioned variable-span transition advances both initial and generated states. We introduce partitioned-path learning, using observed endpoint images to supervise continued predictions from generated states and guide intermediate state formation through subsequent prediction errors. Figure 2 shows the architecture (a) and learning mechanism (b).

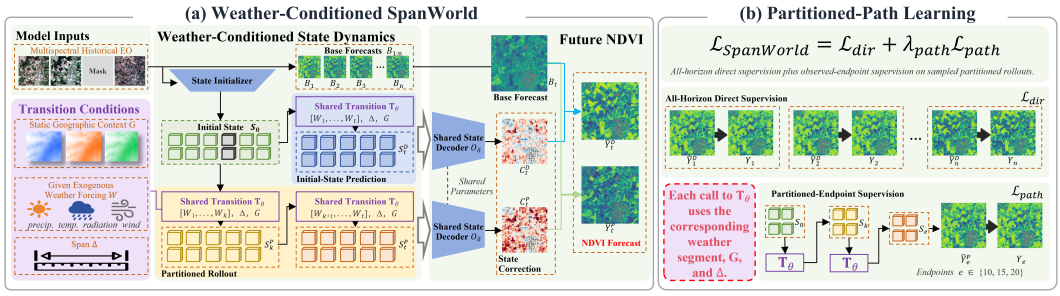


Figure 2: **SpanWorld architecture and partitioned-path learning.** (a) Direct (D) and partitioned (P) routes share weather-conditioned dynamics and a readout module; state-derived image corrections are added to history-derived baseline predictions. (b) Training combines supervision of direct predictions at all forecast horizons with endpoint supervision after sampled partitioned rollouts. Future target images serve only as supervision targets for predictions.

3.1 WEATHER-CONDITIONED VARIABLE-SPAN TRANSITION

To support continued prediction over different time intervals, SpanWorld represents the dynamic state as a spatial grid of image-patch features. A shared transition conditioned on each interval's weather and length reuses states formed by earlier transitions to predict subsequent changes.

Let time 0 denote the end of the history window. The historical context $\mathcal{H}$ comprises incomplete remote-sensing imagery, observation validity masks, and historical weather; G denotes static geographic conditions. Given future weather $W_{1:n}$, the model predicts dense NDVI images $Y_{1:n}$ over the next n five-day steps, without new remote-sensing imagery. The history encoder E_ϕ maps $\mathcal{H}$ and G to future baseline predictions $B_{1:n}$ and an initial dynamic state S_0:

$$(B_{1:n}, S_0) = E_\phi(\mathcal{H}, G). \tag{1}$$

Here, ϕ denotes the encoder parameters, and S_0 is obtained by projecting features at the end of the history window. Both the baseline predictions $B_{1:n}$ and the initial state S_0 are formed without future weather. The baselines retain predictive information from past observations. The shared transition uses future weather to update the dynamic state, whose readout then contributes to the prediction.

To handle intervals of different lengths, we define each transition by its start and end times. Given an input state S_a at time a, the transition produces a state S_b at time b:

$$S_b = T_\theta\left(S_a;\, W_{a+1:b}, G, \Delta\right), \qquad \Delta = b - a, \quad 0 \le a < b \le n. \tag{2}$$

Here, T_θ is the shared state transition with learnable parameters θ, Δ is the length of the current interval, $W_{a+1:b}$ is the weather sequence over that interval in chronological order, and G denotes static geographic conditions. The input S_a can be either the state initialized from history or a state generated by an earlier transition.

A shared GRU encodes the current interval's weather chronologically from a zero hidden state. Its final output and the span embedding are broadcast over the spatial grid, concatenated with geographic features at each location, and fused. These features are concatenated with the normalized current state and passed through a multilayer perceptron (MLP), whose output is added to the input state. The network is shared across spatial locations and intervals. Each transition thus uses its own interval's conditions, while the dynamic state carries information from preceding transitions.

For target time t, the direct path applies the shared transition once to obtain S_t^D from S_0. The partitioned path follows a temporal partition $0 = \tau_0 < \tau_1 < \cdots < \tau_m = t$, with $m \ge 2$, and its states carry superscript P:

$$\begin{aligned} S_t^D &= T_\theta(S_0;\, W_{1:t}, G, t), \\ S_{\tau_0}^P &= S_0, \\ S_{\tau_j}^P &= T_\theta\left(S_{\tau_{j-1}}^P;\, W_{\tau_{j-1}+1:\tau_j}, G, \tau_j - \tau_{j-1}\right), \qquad j = 1, \ldots, m. \end{aligned} \tag{3}$$

The resulting states depend on the chosen partition, with endpoint state $S_t^P = S_{\tau_m}^P$. Both paths connect the same start and end dates under the same ordered weather sequence. After the initial transition, subsequent transitions take the preceding interval's generated state as input, supporting continued prediction through repeated application of the shared dynamics.

A shared readout O_δ maps each location's state features to an image patch. Assembling these patches yields an image-space correction C_t^r, added to the history-derived baseline B_t for the same target date:

$$C_t^r = O_\delta(S_t^r), \qquad \hat{Y}_t^r = B_t + C_t^r, \qquad r \in \{D, P\}. \tag{4}$$

Here, δ denotes the readout parameters, and $\hat{Y}_t^r$ is the complete prediction. Direct and partitioned paths share the baselines, transition parameters, and readout. The rollout path therefore affects the prediction through the dynamic state it produces. Appendix A.1 details the encoder backbone and its source, state initialization, transition, and readout.

A state S_k formed at time k can support predictions at multiple later dates. For each target date $t > k$, the model independently advances the same S_k using the shared transition with weather $W_{k+1:t}$, geographic conditions G, and span $t - k$. The resulting state's readout is added to the target date's baseline B_t, retaining predictive information from history during continued prediction.

3.2 Partitioned-Path Learning

To learn states that support subsequent prediction, SpanWorld supplements supervision of direct predictions with endpoint supervision along partitioned paths. The direct objective constrains predictions over the entire forecast window:

$$\mathcal{L}_{\text{dir}} = \ell_{\text{seq}}\left(\hat{Y}_{1:n}^D, Y_{1:n}\right). \tag{5}$$

Here, $\hat{Y}_{1:n}^D$ is the image sequence obtained by making a separate prediction for each future date from the initial state S_0, and $Y_{1:n}$ contains the corresponding observed images. The sequence mean squared error ℓ_{seq} accounts for observation validity and vegetation masks.

We additionally supervise predictions along partitioned paths using observed endpoint images, training the shared transition to operate on model-generated states. Each endpoint in a predefined set $\mathcal{E}$ is associated with a candidate set of temporal partitions with multiple segments. Each candidate consists of positive integer spans that sum to the endpoint's time index. At each training update, we independently sample one path uniformly from each endpoint's candidate set, thereby selecting its segment count and boundaries. The endpoints and candidate sets are listed in Appendix A.3.

From the same initial state S_0, the model repeatedly applies the shared transition along each sampled path, producing endpoint state S_e^P and complete prediction $\hat{Y}_e^P = B_e + O_\delta(S_e^P)$. The partitioned-path loss is

$$\mathcal{L}_{\text{path}} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \ell_{\text{img}} \left(\hat{Y}_e^P, Y_e \right). \tag{6}$$

Here, Y_e is the observed image at endpoint e, used only as a loss target, and ℓ_{img} is the masked mean squared error over valid vegetation pixels. All states along the path are model-generated. Sampling different temporal partitions trains the shared transition to predict subsequent observations from states formed over different preceding intervals.

Partitioned-path learning places intermediate states in the computational path of subsequent predictions. Gradients propagate through every segment without truncation, allowing endpoint prediction errors to guide the formation of intermediate states through the subsequent transitions. These states are therefore learned to support later state evolution and image prediction.

The joint objective is

$$\mathcal{L}_{\text{SpanWorld}} = \mathcal{L}_{\text{dir}} + \lambda_{\text{path}} \mathcal{L}_{\text{path}}, \tag{7}$$

where λ_{path} controls the relative weight of partitioned-path supervision. Both losses supervise complete image predictions, covering direct predictions from the initial state and subsequent predictions from generated states. Appendix A.2 details masking and normalization, while Appendix A.4 provides the training configuration.

4 Experiments

We test whether partitioned-path learning improves continued prediction from model-generated states when direct supervision already covers all future target dates. Controlled comparisons within the same architecture examine the effect of supervision paths. Evaluations under unseen temporal partitions test the shared transition's endpoint predictions, while component and input interventions examine the contributions of dynamic states and weather.

4.1 Evaluation Setup

We evaluate SpanWorld on the in-distribution (IID), temporal out-of-distribution (OOD-T), spatial out-of-distribution (OOD-S), and joint temporal-spatial out-of-distribution (OOD-ST) splits of GreenEarthNet (Benson et al., 2024). Given a 50-day historical context containing incomplete remote-sensing observations, together with weather for the forecast window, the model predicts NDVI images for the next 100 days at five-day intervals, without additional remote-sensing observations during prediction. We evaluate standard forecasting using the official land-cover-balanced R^2 and root mean squared error (RMSE). For continued prediction, we use mean squared error (MSE) averaged equally across samples, comparing predictions over the same target dates, samples, and valid observations. Standard forecasting comparisons are provided in Appendix B. We focus below on the subsequent use of model-generated states and how supervision paths affect this capability.

4.2 Continued Prediction from Generated States

To examine the effect of the supervision path, we compare SpanWorld with a control that applies the additional endpoint supervision to direct predictions (the direct-endpoint control). Both models use the same architecture and initialization, are trained for the same number of updates, receive supervision of direct predictions at all forecast horizons, and use the same endpoint images for additional supervision. The difference is whether this additional supervision is applied to predictions

obtained through partitioned or direct rollouts. At evaluation, each model reaches day 50 through one of three paths: a single 50-day segment, 25+25 days, or 15+35 days. The 15+35-day partition is unseen during training. Each model then predicts the NDVI image at each target date from day 55 to day 100 independently from its own fixed day-50 state, using weather over the corresponding interval, geographic conditions, and the history-derived baseline prediction. All comparisons use the same 2,837 IID samples and subsequent target window.

The two supervision schemes yield similar direct prediction errors, but their performance differs markedly when generated states are reused (Table 1). When predicting the day-55–100 targets directly from the initial state, the control and SpanWorld achieve MSEs of 0.041480 and 0.041285, respectively. Across all three rollout paths, the control's continued prediction error increases, whereas SpanWorld's error remains close to its own direct prediction error. For the 15+35-day partition, unseen during training, SpanWorld achieves an MSE of 0.041448, a 10.96% reduction from the control's 0.046550. With the same target-date coverage and endpoint images for additional supervision, partitioned-path learning reduces degradation in continued prediction. The additional endpoint images already fall within the dates covered by direct supervision. The improvement therefore concerns how those observations constrain the state-transition path: the endpoint error also evaluates predictions that reuse a generated intermediate state. This comparison shows that supervising the subsequent use of generated states helps preserve predictive capability after rollout.

Table 1: **Direct and continued prediction over a shared subsequent window.** Predictions are evaluated on the same 2,837 IID samples for days 55–100. Predictions start from the initial state or each model's fixed day-50 state reached along the indicated path. MSE is averaged equally across samples. Differences are SpanWorld minus control. Their 95% intervals are computed by paired geographic-cluster bootstrap resampling with trained models held fixed. Differences and relative improvements use unrounded values. The 15+35-day partition is unseen during training. Calculation details are in Appendix C.

Starting state	Control	SpanWorld	Δ MSE	95% CI
Initial state	0.041480	**0.041285**	-0.000195	$[-0.000382, -0.000015]$
After 50 days	0.043713	**0.041688**	-0.002025	$[-0.002680, -0.001300]$
After 25+25 days	0.046488	**0.041536**	-0.004952	$[-0.005948, -0.003868]$
After 15+35 days	0.046550	**0.041448**	-0.005101	$[-0.006107, -0.004009]$

* Within each row, the lower MSE is bolded.

The spatial predictions illustrate this difference (Figure 3). In the example shown, the two schemes produce similar direct predictions. When predicting day 100 after a 15+35-day rollout, the control overpredicts NDVI over a substantial area and exhibits increased absolute error, whereas SpanWorld's prediction pattern and error level change less. This case was selected from the available visualizations based on the difference in degradation after rollout (Appendix G). It illustrates how the improvement appears spatially. Table 1 provides aggregate evidence over the shared sample set.

4.3 Endpoint Prediction under Unseen Temporal Partitions

Using the shared transition in partitioned rollouts also requires it to accommodate temporal partitions unseen during training. We therefore examine whether endpoint predictions remain accurate when the number and lengths of rollout segments change. We compare SpanWorld's predictions under nine temporal partitions unseen during training (Appendix D) with direct predictions at the same endpoints: days 50, 75, and 100 (Figure 4(a)). IID sample eligibility is determined by valid observations at each endpoint, and MSE is pooled over valid pixels. Across these partitions, the ratio of partitioned to direct prediction MSE ranges from 0.992 to 1.023, indicating endpoint accuracy close to direct prediction. This result shows that the shared transition's predictive capability extends to the evaluated temporal partitions unseen during training. Together with the continued-prediction comparison, it addresses two stages of partitioned rollout: producing an accurate image at the reached endpoint and using the resulting state to predict later observations. The former examines the composition of variable-span transitions, while the latter evaluates the subsequent use of the states they generate.

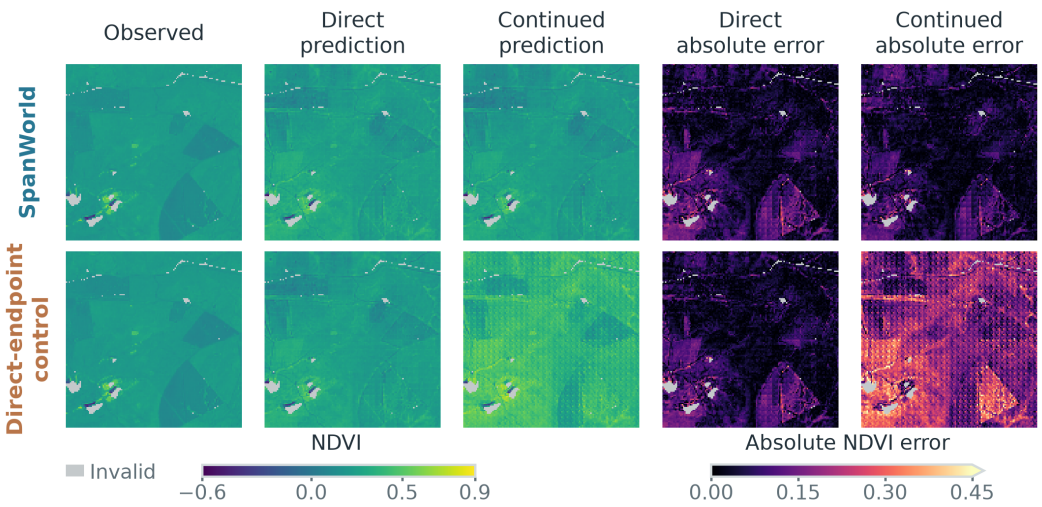


Figure 3: **Direct and continued day-100 predictions for a high-improvement IID example.** Continued predictions use each model's state after a 15+35-day rollout. NDVI panels share one color scale, and absolute-error panels share another. A common mask applies throughout, with invalid observations in gray. Errors above 0.45 saturate for display only.

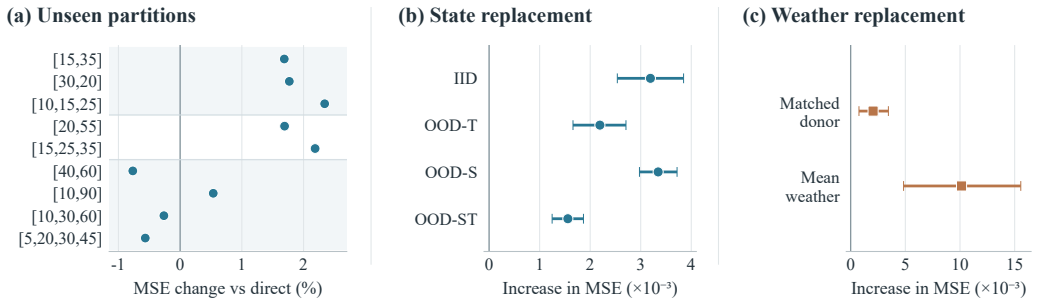


Figure 4: **Unseen temporal partitions and input replacements.** (a) Percentage change in pooled endpoint MSE versus direct prediction at the same endpoint on IID. Brackets show segment lengths (days). Groups reach days 50/75/100, top to bottom. (b) State-replacement increase in sample-averaged MSE for days 55–100. (c) Weather-replacement MSE increase over the full 100-day window on 84 hot-dry OOD-T samples. T/S/ST denote temporal, spatial, and joint distribution shifts. Mean weather denotes normalized-mean weather. In (b,c), positive values favor the original state or actual weather, and error bars show paired geographic-cluster bootstrap 95% confidence intervals.

4.4 Contributions of State Evolution and Weather

To examine the dynamic contribution beyond the history-derived baseline, we hold the trained model fixed and either retain only the baseline prediction or replace the shared transition with an identity mapping, keeping the state at its initial value. Both interventions increase RMSE on all four splits (Table 2). On IID, RMSE rises from 0.1555 for the full model to 0.1708 with only the history-derived baseline and to 0.2150 with the identity transition. The two interventions examine complementary aspects of the dynamic branch. Retaining only the baseline removes the state-derived image correction, whereas the identity transition preserves the readout of the initial state but omits its evolution under the interval's weather. These results show that both the state correction and its dynamic update contribute to the full model's predictions, with the dynamic branch providing useful predictive information beyond the history-derived baseline.

Continued prediction uses both a generated state and a history-derived baseline prediction. We therefore examine whether the generated state provides useful predictive information in the presence of that baseline, and whether this contribution depends on matching the state to the current sample.

Table 2: **Predictive contributions of state correction and dynamic transition.** Official RMSE over the full 100-day forecast window under inference-time interventions on the same trained model. Full metrics and intervention details are provided in Appendix E.

Setting	IID	OOD-T	OOD-S	OOD-ST
Full model	**0.1555**	**0.1509**	**0.1621**	**0.1542**
History baseline only	<u>0.1708</u>	<u>0.1627</u>	<u>0.1755</u>	<u>0.1650</u>
Identity transition	0.2150	0.2150	0.2196	0.2164

* Best results are bolded, and second-best results are underlined.

We replace each sample's generated state with one from another sample matched on historical NDVI features and land-cover composition, while retaining the original sample's history-derived baseline prediction, geographic conditions, and subsequent weather. We evaluate predictions for days 55–100 (Figure 4(b)). State replacement increases mean MSE on all four splits, with the 95% confidence intervals for these increases lying entirely above zero. Thus, even with the baseline prediction unchanged, replacing the state raises subsequent prediction error, indicating that the sample's own generated state contributes to continued prediction.

Weather replacement further tests whether using each sample's actual future weather improves prediction. On 84 hot-dry samples from the OOD-T split (Appendix F), we hold the model, historical observations, and evaluation targets fixed and replace actual weather with either the weather sequence of a matched sample with normal weather or normalized-mean weather. We evaluate the full 100-day prediction window (Figure 4(c)). Both replacements increase mean MSE, with a larger increase under mean-weather conditions. On this evaluated subset, using the actual weather associated with each sample lowers prediction error, indicating that weather conditioning provides useful predictive information. The replacement changes the forcing supplied to the shared transition while retaining the same observed targets. The resulting error increase links weather conditioning to the prediction of actual future vegetation, complementing the evidence that generated states themselves remain useful.

Together, these results show that supervision paths affect continued prediction from generated states even when direct-prediction errors are similar. Partitioned-path learning applies observation-based supervision to predictions made after generated states are reused, reducing degradation across the evaluated paths. Unseen partitions, component interventions, and input replacements further characterize the use of the shared transition and the contributions of dynamic states and weather.

5 Conclusion

We present SpanWorld, a variable-span world model that forecasts vegetation change from incomplete remote-sensing observations given future weather. Its shared state transition supports direct prediction and partitioned rollout. Alongside supervision of direct predictions at all forecast horizons, partitioned-path learning uses observed endpoint images to supervise subsequent predictions from model-generated intermediate states. In controlled comparisons with the architecture, initialization, and endpoint images used for supervision held fixed, partitioned-path supervision reduces degradation in continued prediction across the evaluated paths, while the two training approaches yield similar direct-prediction errors.

These findings show that, even when supervision of direct predictions covers all future target dates, the rollout path along which supervision is applied affects subsequent prediction from model-generated states. For world models intended for continued rollout, learning and evaluation should consider both prediction accuracy from observations and the ability of model-generated representations to support further prediction of actual future change.

AI USE STATEMENT

We used generative AI tools to assist with the implementation of methods and evaluation code and with Chinese-to-English translation, and to provide feedback on methodology, experiments, and the authors' interpretation of experimental results. Additional uses included manuscript drafting and language editing, literature search and synthesis, and preparation of figures, tables, and LaTeX files. The research hypotheses and the ideas guiding their revision originated with the authors. The authors reviewed all AI-assisted work, including the code, references, experimental results, and manuscript, and take responsibility for the final content of this paper.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Advances in Neural Information Processing Systems*, 2024.

Mahmoud Assran et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

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*, 2015.

Vitus Benson, Claire Robin, Christian Requena-Mesa, Lazaro Alonso, Nuno Carvalhais, José Cortés, Zhihan Gao, Nora Linscheid, Mélanie Weynants, and Markus Reichstein. Multi-modal learning for geospatial vegetation forecasting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. In *International Conference on Learning Representations*, 2022.

Maxime Burchi and Radu Timofte. Learning transformer-based world models with contrastive predictive coding. In *International Conference on Learning Representations*, 2025.

Junfeng Chen and Kailiang Wu. Deep-OSG: Deep learning of operators in semigroup. *arXiv preprint arXiv:2302.03358*, 2023.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, 2018.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, 2018.

Fei Deng, Ingook Jang, and Sungjin Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, 2022.

Codruţ-Andrei Diaconu, Sudipan Saha, Stephan Günnemann, and Xiao Xiang Zhu. Understanding the role of weather data for earth surface forecasting using a convlstm-based model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022.

Zhangyang Gao, Cheng Tan, Lirong Wu, and Stan Z. Li. SimVP: Simpler yet better video prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022a.

Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Yuyang Bernie Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring space-time transformers for earth system forecasting. In *Advances in Neural Information Processing Systems*, 2022b.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, 2019.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

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*, 2019.

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

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, 2022.

Irene Iele, Elena Mulero Ayllón, Paolo Soda, and Matteo Tortora. Vegsim: A geospatial world model for scenario-conditioned vegetation simulation. *arXiv preprint arXiv:2606.21961*, 2026.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henry Michalewski. Model-based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020.

Alex M. Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C. Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems*, 2016.

Vincent Le Guen and Nicolas Thome. Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural pde solvers. In *Advances in Neural Information Processing Systems*, 2023.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations*, 2023.

Christian Requena-Mesa, Vitus Benson, Markus Reichstein, Jakob Runge, and Joachim Denzler. Earthnet2021: A large-scale dataset and challenge for earth surface forecasting as a guided video prediction task. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2021.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. In *International Conference on Learning Representations*, 2023.

Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems*, 2019.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, 2020.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588: 604–609, 2020.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, 2015.

Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991. doi: 10.1145/122344.122377.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD: Masked conditional video diffusion for prediction, generation, and interpolation. In *Advances in Neural Information Processing Systems*, 2022.

Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S. Yu. PredRNN: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Advances in Neural Information Processing Systems*, 2017.

Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. PredRNN++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, 2018.

Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*, 2015.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. In *Advances in Neural Information Processing Systems*, 2021.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. STORM: Efficient stochastic transformer based world models for reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Sijie Zhao, Hao Chen, Xueliang Zhang, Pengfeng Xiao, Lei Bai, and Wanli Ouyang. VegeDiff: Latent diffusion model for geospatial vegetation forecasting. *IEEE Transactions on Geoscience and Remote Sensing*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, 2025.

A Method and Implementation Details

A.1 Inputs and Network Implementation

The model uses 10 historical steps to forecast the next 20 steps at five-day intervals and a spatial size of 128×128 pixels. Remote-sensing inputs comprise NDVI, B02, B03, B04, and B8A sampled at the end of each five-day interval; the output is NDVI. Preprocessing follows the Contextformer data adapter (`earthnet-models-pytorch v0.1.0`). Eight weather variables, ordered as `fg, hu, pp, qq, rr, tg, tn, tx`, are standardized using fixed statistics. Their five-day means, minima, and maxima are concatenated in this order to form 24-dimensional weather inputs. Geographic inputs comprise NASA, ALOS, and Copernicus DEM channels, each divided by 500; land cover is used only for supervision masks. Missing input values are filled with zero. Observation masks combine cloud masks and scene classifications, with missing mask values treated as invalid.

The history encoder follows the Contextformer 6M configuration (Benson et al., 2024). A single forward pass produces the baseline sequence $B_{1:n}$ and features for state initialization. During this pass, future image positions are masked and future weather is zeroed, so neither $B_{1:n}$ nor S_0 receives future weather. The actual future weather enters the subsequent state transitions separately. A historical image-patch embedding is masked if any pixel in the patch has a positive mask value. Features at the last historical step from the final Transformer block are projected to obtain S_0. The readout maps each state vector to a 4×4 image patch, assembles them into an image-space correction, and adds it to B_t at the absolute target date, with a fixed correction coefficient of 1.

Each weather interval $W_{a+1:b}$ is encoded in temporal order from a zero GRU hidden state, using the final output as its representation. The span encoding uses the local interval length $b-a$. Weather and span representations are broadcast over the spatial grid and fused with location-specific geographic features. The fused condition and normalized state enter an MLP that predicts an increment to be added to the input state. The transition applies a shared network at each spatial location without an additional cross-location mixing layer; parameters are shared across forecast horizons and partitions.

For all-horizon direct prediction, a single GRU pass over $W_{1:n}$ produces weather representations for every prefix, equivalent to encoding each prefix from a zero hidden state. The horizon-specific dynamic states are then computed independently from the same S_0 in a batch. In partitioned rollout, the weather GRU starts from a zero hidden state for each segment, while the dynamic state is passed between successive segments.

Table A1: Input and network specifications. Shapes omit the batch dimension. MLP widths are listed in order, with GELU after intermediate hidden layers.

Component	Configuration
Historical imagery; validity mask	$10 \times 5 \times 128 \times 128$; $10 \times 1 \times 128 \times 128$
Historical weather; future weather	10×24; 20×24
Geographic inputs; forecast output	$3 \times 128 \times 128$; $20 \times 1 \times 128 \times 128$
History backbone	Contextformer 6M with PVT spatial encoding; temporal Transformer depth 3, width 256, 8 attention heads, MLP expansion ratio 4
State grid	$32 \times 32 \times 256$, corresponding to 4×4 image patches; batch and spatial dimensions are flattened in the implementation
State projection	LayerNorm $\rightarrow$ Linear $256 \rightarrow 512 \rightarrow$ GELU $\rightarrow$ Linear $512 \rightarrow 256 \rightarrow$ LayerNorm
Weather encoder	Single-layer GRU; input dimension 24, hidden dimension 128
Geographic encoder	4×4 average pooling; MLP $3 \rightarrow 64 \rightarrow 64$
Span encoding	Fixed 64-dimensional sine/cosine encoding of the span in five-day steps
Condition fusion	Concatenate $128 + 64 + 64$ condition dimensions; MLP $256 \rightarrow 256 \rightarrow 256$

Table A1 (continued)

Component	Configuration
Residual state update	Concatenate the LayerNorm-normalized state and fused condition; MLP $512 \to 512 \to 512 \to 256$, added to the input state
Prediction correction readout	Linear $256 \to 16$, followed by spatial reassembly

A.2 Loss Computation

Let i, t, p index samples, forecast steps, and pixels. The observation indicator c_{itp} equals 1 when the observation-mask value is below 1, and the vegetation indicator v_{ip} equals 1 for land-cover codes in $[10, 40]$. We use $\epsilon = 10^{-8}$. Direct predictions use the inherited sequence loss, which first averages over valid dates at each pixel and then aggregates across pixels and samples. Let $q_{ip} = \max_t \mathbf{1}[\widehat{Y}^D_{itp} \neq -1]$, where -1 is the inherited invalid-prediction marker:

$$\ell_{\text{seq}} = \frac{\displaystyle\sum_{i,p} v_{ip} q_{ip} \frac{\sum_{t=1}^{n} c_{itp} (\widehat{Y}^D_{itp} - Y_{itp})^2}{\sum_{t=1}^{n} c_{itp} + \epsilon}}{\displaystyle\sum_{i,p} v_{ip} q_{ip} + \epsilon}. \tag{8}$$

The endpoint image loss instead aggregates directly over valid vegetation pixels at the target date:

$$\ell_{\text{img}}(\widehat{Y}_e, Y_e) = \frac{\displaystyle\sum_{i,p} v_{ip} c_{iep} (\widehat{Y}_{iep} - Y_{iep})^2}{\displaystyle\sum_{i,p} v_{ip} c_{iep} + \epsilon}. \tag{9}$$

Partitioned-path supervision averages the image losses equally over three endpoints. Both supervision terms have unit weight, giving $\lambda_{\text{path}} = 1$, and supervise the full prediction obtained by adding the state correction to B_t. The normalization above is applied within each device's microbatch; losses are scaled by the gradient-accumulation factor, and gradients are averaged across devices. No target-standard-deviation scaling or reweighting is applied. A pixel with no valid observations throughout the target window contributes zero sequence error but remains in the outer denominator if it satisfies the outer mask. An endpoint with no valid supervision pixels contributes zero loss and remains in the three-endpoint average.

A.3 Partition Sampling and Training Procedure

The training endpoint set is $\mathcal{E} = \{10, 15, 20\}$, corresponding to days 50, 75, and 100. Each update independently samples one multi-segment path uniformly from each endpoint's candidate set. The plan is determined by the training seed and update index and shared across samples, microbatches, and devices within that update. Each candidate partition is written as a sequence of segment lengths in five-day steps. The lengths sum to its endpoint index. Sampling is restricted to these candidates. Unseen evaluation partitions are listed separately in D.

Table A2: Training candidates for partitioned-path supervision.

Endpoint	Forecast lead time	Candidate segment lengths
10	50 days	$(5, 5)$, $(5, 3, 2)$
15	75 days	$(7, 8)$, $(7, 4, 4)$
20	100 days	$(10, 10)$, $(10, 5, 5)$, $(5, 5, 5, 5)$

After one history-encoding pass, all-horizon direct predictions and the three partitioned endpoint predictions are computed from the same S_0. Each partitioned path decodes an image only at its final

endpoint. In the current training stage, observed target imagery enters only the loss, whose gradients propagate through all segments of each sampled path. The direct-endpoint control retains the model, endpoints, losses, and optimization settings. It performs the same random draws, then replaces each sampled path with a single direct transition from S_0 for the additional endpoint supervision.

Continuation evaluation fixes the same generated state S_k as the starting point for each subsequent target date, using $W_{k+1:t}$, the local span $t-k$, the original sample's geography, and the retained B_t. For each target date, a separate call to the shared transition advances the same fixed state S_k. The prediction combines the resulting state's readout with the retained baseline B_t. For partitioned-path training, each supervised endpoint is reached along its own sampled path from S_0. Evaluation dates and statistics are specified in C.

A.4 Initialization and Optimization

The model and direct-endpoint control inherit the same full checkpoint. Its immediate preceding training stage started from an existing forecasting model and ran for 14,880 updates, using observed NDVI supervision, prediction distillation from a frozen weather-conditioned teacher, and terminal-state supervision with targets encoded from observed future imagery by a frozen encoder. The first two terms had weights of 1 and 0.5. State supervision used cosine distance after LayerNorm, with its weight increasing linearly from 0 to 0.02 during the first 20% of updates, remaining at 0.02 until 80%, and changing to 0.01 for the final 20%. The history backbone was frozen for the first 80%, after which its final temporal Transformer block was unfrozen.

The current stage starts with a new optimizer and schedule and uses only the two image-supervision terms in A.2, without distillation or future-state targets. Only the final temporal Transformer block of the history backbone is finetuned; the state projection, condition encoders, fusion, transition, and readout modules are trained. History encoding runs in evaluation mode to disable random temporal masking while retaining gradients for unfrozen parameters.

Table A3: Optimization settings for the current training stage.

Setting	Value
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$
Learning rate for state and condition modules	3×10^{-5}
Learning rate for the unfrozen history-backbone block	9.9×10^{-7}, scaling factor 0.033
Weight decay	0
Learning-rate schedule	Linear warmup for 300 updates, followed by cosine decay to 0
Batch size and gradient accumulation	Global batch 64; 4 devices $\times$ 8 samples; accumulation 2
Gradient-norm clipping threshold	1
Training duration	40 epochs, 14,880 updates
Random seed	42

Reported results use the prespecified checkpoint at update 14,880 rather than the checkpoint with the best validation score. Each model in the core supervision-path comparison uses one training run with seed 42. Evaluations of unseen partitions and interventions use the corresponding fixed checkpoint.

B STANDARD FORECASTING: SETUP AND FULL RESULTS

B.1 DATA AND EVALUATION PROTOCOL

We use the four chopped GreenEarthNet splits (Benson et al., 2024), with manifest sizes listed in Table B1. Preprocessing follows Appendix A.1. All evaluations use 10 historical and 20 future five-day steps. Future remote-sensing images and their validity masks are used only for evaluation.

The official scorer computes temporal pixel metrics and aggregates them within cube–land-cover groups, across cubes within each class, and then across land-cover classes. Eligibility requires the cloud/scene mask, a land-cover code below 41, positive minimum valid NDVI, at least 10 valid future observations and three outside the forecast window, and target temporal standard deviation above 0.1. R^2_{LC} averages squared correlations rather than $1 - \mathrm{SSE}/\mathrm{SST}$; RMSE is aggregated after taking square roots. NSE is transformed to normalized NSE before aggregation and inverted afterward. $|\mathrm{Bias}|$ is absolute bias, and RMSE25 is the RMSE over the first 25 forecast days. These metrics are distinct from the sample-averaged MSE in Appendix C.

All learned methods use seeds 27/42/97. The four published learning baselines use the GreenEarthNet authors' released checkpoints (Zenodo record 10793870), re-evaluated with shared manifests, input versions, masks, and scoring code. Our weather-conditioned ConvGRU uses one checkpoint per seed across all four splits. Persistence is deterministic. In Table B2, $\pm$ denotes the sample standard deviation across three seeds. All methods share the evaluation protocol, with training budgets and initialization set according to their respective procedures. Table B1 gives manifest sample counts; endpoint-eligible counts are reported for each experiment. OOD-S uses the repaired data version from the formal standard evaluation.

Table B1: Standard forecasting manifests.

Split	Manifest	Samples
IID	iid_chopped	2,856
OOD-T	ood-t_chopped	1,904
OOD-S	ood-s_chopped	10,536
OOD-ST	ood-st_chopped	7,024

The weather-conditioned ConvGRU is our recurrent baseline implementation. Two 64-channel convolutional layers encode historical five-band imagery, validity masks, and three DEM channels. A 96-channel ConvGRU operates on a 128×128 grid with 24-dimensional weather and geography, advancing one five-day step per call. Forecasting feeds back the previous NDVI prediction, with readout $\mathrm{sigmoid}(\mathrm{Conv}_{1\times1}(h_t) + \widehat{Y}_{t-1})$. The previous NDVI prediction is added to the convolutional output before the sigmoid, so both terms pass through the nonlinear readout. Inference proceeds without future remote-sensing observations.

B.2 FULL METRICS AND SEED RESULTS

Tables B2–B4 report full metrics, per-seed primary scores for all six learned methods, and three-seed ConvGRU forecast-window RMSE. Released-baseline run records retain four decimals, from which their means and sample standard deviations are calculated. Each SpanWorld seed uses one checkpoint across all splits. Its R^2_{LC} is below PredRNN and Contextformer on every split. Relative to ConvGRU, SpanWorld has higher R^2_{LC} and lower RMSE on IID and OOD-T; on OOD-S, it has higher R^2_{LC} but slightly higher RMSE; on OOD-ST, it has slightly lower R^2_{LC} and lower RMSE.

Table B2: Full standard forecasting metrics on four splits. Learned methods report three-seed means ± sample standard deviations; deterministic results are point estimates. ConvGRU denotes the weather-conditioned recurrent baseline.

Method	Seeds	R^2 ↑	RMSE ↓	NSE ↑	|Bias| ↓	RMSE25 ↓
			IID			
SpanWorld	3	0.5221 ± 6.49e-5	0.1555 ± 1.42e-6	-0.1370 ± 9.51e-5	0.0997 ± 4.00e-6	0.0912 ± 1.75e-5
ConvGRU	3	0.5195 ± 0.0023	0.1568 ± 0.0021	-0.1498 ± 0.0217	0.0989 ± 0.0028	0.0937 ± 6.91e-5
Persistence	–	0.0000	0.2213	-1.1716	0.1610	0.1028
ConvLSTM 1M	3	0.5107 ± 0.00150	0.1557 ± 0.00110	-0.1046 ± 0.0115	0.0978 ± 0.00121	0.1005 ± 0.00168
PredRNN 1M	3	**0.5414** ± 0.0110	**0.1423** ± 0.00450	**0.0427** ± 0.0481	**0.0869** ± 0.00365	**0.0888** ± 0.00253
SimVP 6M	3	0.4988 ± 0.00609	<u>0.1461</u> ± 0.00171	<u>-0.0191</u> ± 0.0227	<u>0.0871</u> ± 0.00150	0.1009 ± 0.00272
Contextformer 6M	3	<u>0.5333</u> ± 6.11e-4	0.1484 ± 9.02e-4	-0.0387 ± 0.0131	0.0941 ± 8.50e-4	<u>0.0893</u> ± 4.58e-4
			OOD-T			
SpanWorld	3	0.5726 ± 1.08e-4	0.1510 ± 2.74e-5	-0.1068 ± 4.31e-4	0.1014 ± 3.21e-5	<u>0.0824</u> ± 2.03e-5
ConvGRU	3	0.5683 ± 0.0012	0.1532 ± 0.0011	-0.1326 ± 0.0091	0.1014 ± 0.0010	0.0872 ± 0.0017
Persistence	–	0.0000	0.2157	-1.2248	0.1583	0.0850
ConvLSTM 1M	3	0.5483 ± 0.00696	0.1615 ± 0.00182	-0.1899 ± 0.0130	0.1097 ± 0.00228	0.1022 ± 0.00272
PredRNN 1M	3	**0.5924** ± 0.00153	<u>0.1474</u> ± 0.00212	<u>-0.0375</u> ± 0.0266	<u>0.0974</u> ± 0.00470	0.1018 ± 0.00192
SimVP 6M	3	0.5616 ± 0.00532	0.1492 ± 0.00191	-0.0660 ± 0.0165	0.0977 ± 0.00202	0.1043 ± 0.00161
Contextformer 6M	3	<u>0.5877</u> ± 0.00436	**0.1431** ± 8.19e-4	**0.0051** ± 0.0168	**0.0936** ± 5.03e-4	**0.0818** ± 0.00315
			OOD-S			
SpanWorld	3	0.4949 ± 2.88e-5	0.1621 ± 9.59e-6	-0.1763 ± 1.26e-4	0.1018 ± 1.37e-5	0.0970 ± 1.55e-5
ConvGRU	3	0.4830 ± 0.0015	0.1617 ± 0.0023	-0.1695 ± 0.0302	0.0991 ± 0.0031	0.0998 ± 0.0007
Persistence	–	0.0000	0.2258	-1.1336	0.1625	0.1127
ConvLSTM 1M	3	0.4761 ± 0.00373	0.1622 ± 0.00105	-0.1540 ± 0.0113	0.0998 ± 0.00105	0.1038 ± 0.00176
PredRNN 1M	3	**0.5087** ± 0.00926	**0.1494** ± 0.00429	**-0.0103** ± 0.0500	<u>0.0888</u> ± 0.00435	**0.0935** ± 0.00337
SimVP 6M	3	0.4649 ± 0.00635	<u>0.1529</u> ± 0.00125	<u>-0.0703</u> ± 0.0184	**0.0882** ± 0.00164	0.1069 ± 0.00351
Contextformer 6M	3	<u>0.5021</u> ± 0.00197	0.1549 ± 0.00160	-0.0790 ± 0.0221	0.0960 ± 0.00166	<u>0.0948</u> ± 6.56e-4
			OOD-ST			
SpanWorld	3	0.5407 ± 1.88e-4	0.1542 ± 2.51e-5	-0.1551 ± 4.30e-4	0.1016 ± 1.86e-5	<u>0.0871</u> ± 2.38e-5
ConvGRU	3	0.5415 ± 0.0033	0.1553 ± 0.0021	-0.1710 ± 0.0226	0.1008 ± 0.0018	0.0924 ± 0.0008
Persistence	–	0.0000	0.2183	-1.2401	0.1607	0.0966
ConvLSTM 1M	3	0.5234 ± 0.00321	0.1610 ± 9.24e-4	-0.2261 ± 0.0113	0.1059 ± 0.00176	0.1047 ± 0.00204

Table B2 (continued)

Method	Seeds	R^2 ↑	RMSE ↓	NSE ↑	|Bias| ↓	RMSE25 ↓
PredRNN 1M	3	**0.5635** ± 0.00166	<u>0.1478</u> ± 0.00133	<u>-0.0708</u> ± 0.0104	**0.0943** ± 0.00333	0.1045 ± 0.00250
SimVP 6M	3	0.5275 ± 0.00211	0.1553 ± 0.00201	-0.1552 ± 0.0205	0.0999 ± 9.50e-4	0.1082 ± 0.00189
Contextformer 6M	3	<u>0.5567</u> ± 0.00130	**0.1473** ± 0.00104	**-0.0578** ± 0.0154	<u>0.0958</u> ± 0.00137	**0.0860** ± 0.00236

* Within each split and metric, best means are bolded and second-best means are underlined.

Table B3: Per-run primary metrics for all three-seed methods. Each cell gives R^2_{LC} / $\mathrm{RMSE}_{\mathrm{LC}}$. Each seed uses one checkpoint across all splits. Released-baseline run records retain four decimals; their means and standard deviations are computed from those records.

Split	Method	Seed 27	Seed 42	Seed 97
IID	SpanWorld	0.522123 / 0.155531	0.522020 / 0.155529	0.522139 / 0.155532
IID	Weather-conditioned ConvGRU	0.516884 / 0.159180	0.521159 / 0.155214	0.520592 / 0.156141
IID	ConvLSTM 1M	0.5108 / 0.1546	0.5121 / 0.1568	0.5091 / 0.1557
IID	PredRNN 1M	**0.5541** / **0.1371**	**0.5345** / **0.1449**	**0.5357** / **0.1449**
IID	SimVP 6M	0.5003 / <u>0.1443</u>	0.5040 / <u>0.1463</u>	0.4921 / 0.1477
IID	Contextformer 6M	<u>0.5332</u> / 0.1485	<u>0.5340</u> / 0.1493	<u>0.5328</u> / <u>0.1475</u>
OOD-T	SpanWorld	0.572471 / 0.150988	0.572600 / 0.150940	0.572686 / 0.150942
OOD-T	Weather-conditioned ConvGRU	0.568206 / 0.154415	0.567111 / 0.152880	0.569438 / 0.152356
OOD-T	ConvLSTM 1M	0.5466 / 0.1603	0.5560 / 0.1636	0.5424 / 0.1606
OOD-T	PredRNN 1M	**0.5942** / <u>0.1498</u>	**0.5915** / <u>0.1466</u>	**0.5916** / <u>0.1458</u>
OOD-T	SimVP 6M	0.5555 / 0.1510	0.5650 / 0.1472	0.5644 / 0.1495
OOD-T	Contextformer 6M	<u>0.5897</u> / **0.1438**	<u>0.5827</u> / **0.1433**	<u>0.5907</u> / **0.1422**
OOD-S	SpanWorld	0.494926 / 0.162140	0.494875 / 0.162130	0.494923 / 0.162149
OOD-S	Weather-conditioned ConvGRU	0.481694 / 0.164263	0.482702 / 0.159806	0.484653 / 0.161030
OOD-S	ConvLSTM 1M	0.4787 / 0.1611	0.4777 / 0.1632	0.4718 / 0.1622
OOD-S	PredRNN 1M	**0.5187** / **0.1445**	**0.5071** / **0.1526**	**0.5004** / **0.1510**
OOD-S	SimVP 6M	0.4631 / <u>0.1517</u>	0.4720 / <u>0.1529</u>	0.4597 / 0.1542
OOD-S	Contextformer 6M	<u>0.5039</u> / 0.1548	<u>0.5024</u> / 0.1566	<u>0.5000</u> / <u>0.1534</u>
OOD-ST	SpanWorld	0.540543 / 0.154206	0.540760 / 0.154161	0.540918 / 0.154164
OOD-ST	Weather-conditioned ConvGRU	0.538476 / 0.157648	0.541003 / 0.154569	0.545051 / 0.153632
OOD-ST	ConvLSTM 1M	0.5203 / 0.1605	0.5267 / 0.1621	0.5231 / 0.1605
OOD-ST	PredRNN 1M	**0.5647** / <u>0.1485</u>	**0.5616** / <u>0.1487</u>	**0.5642** / **0.1463**
OOD-ST	SimVP 6M	0.5272 / 0.1547	0.5255 / 0.1536	0.5297 / 0.1575
OOD-ST	Contextformer 6M	<u>0.5575</u> / **0.1467**	<u>0.5574</u> / **0.1485**	<u>0.5552</u> / <u>0.1467</u>

* Within each split, seed, and metric, best results are bolded and second-best results are underlined.

Table B4: ConvGRU RMSE by forecast window. Means ± sample standard deviations over seeds 27/42/97; official land-cover-balanced aggregation; each column covers five forecast dates.

Split	Days 5–25	Days 30–50	Days 55–75	Days 80–100
IID	0.0937 ± 6.91e-5	0.1330 ± 0.0014	0.1629 ± 0.0028	0.1772 ± 0.0025
OOD-T	0.0872 ± 0.0017	0.1285 ± 0.0030	0.1608 ± 0.0019	0.1752 ± 0.0038
OOD-S	0.0998 ± 0.0007	0.1371 ± 0.0010	0.1653 ± 0.0031	0.1810 ± 0.0031
OOD-ST	0.0924 ± 0.0008	0.1320 ± 0.0011	0.1600 ± 0.0031	0.1757 ± 0.0040

C Matched Continuation: Protocol and Full Results

C.1 Comparison Conditions and Prediction Routes

SpanWorld and the direct-endpoint control use the same architecture, initialization, supervision of direct predictions at all forecast horizons, and number of updates. Both use the same endpoint images for additional supervision, applied to partitioned and direct predictions, respectively. Appendices A.3–A.4 specify training. Training is matched by update count, with transition-call counts and computational costs determined by the respective rollout paths.

Direct predictions query each date from days 55–100 from S_0. Continued predictions first form S_{10} through 50, 25+25, or 15+35 days of rollout, then query each target independently from that fixed state. Queries use the corresponding interval weather, receiver geography, and retained baseline B_t, without new remote-sensing observations or predicted-image feedback. The 15+35-day partition is absent from the training candidates. An index step spans five days.

C.2 Samples, Estimands, and Uncertainty

A receiver denotes one manifest sample, defined by its historical context and forecast window. Full-suffix eligibility requires all ten horizon records and a positive total valid-pixel count. This yields 2,837 of the 2,856 IID samples; the remaining 19 have no valid suffix pixels. Validity requires an observation-mask value below 1 and land-cover codes in $[10, 40]$, held fixed across models and routes.

Let e_{ik} and v_{ik} denote the sum of squared errors and the valid-pixel count for receiver i at subsequent horizon k. The primary metric is

$$L_i = \frac{\sum_{k=1}^{10} e_{ik}}{\sum_{k=1}^{10} v_{ik}}, \qquad \overline{L} = \frac{1}{N} \sum_{i=1}^{N} L_i. \tag{10}$$

For each receiver, L_i divides the sum of squared errors over the ten horizons by their total valid-pixel count; $\overline{L}$ then averages receivers equally, as defined above. This differs from pooling all receivers' pixels. Paired bootstrap resampling draws 80 Sentinel-2 geographic groups with replacement, retains all within-group receivers and model/route pairings, and recomputes the same sample-averaged statistic. We use 2,000 resamples, random seed 20260914, and percentile 95% intervals. These marginal intervals describe geographic sampling variability with the trained models held fixed. Day-100-only evaluation uses its own validity criterion and 1,670 eligible receivers.

C.3 Full-Suffix, Degradation, and Single-Endpoint Results

Table C1 reports absolute errors and between-model differences for the full suffix; the relative reduction after 15+35 days is 10.96% using unrounded means. Table C2 reports error increases relative to each model's own direct prediction. SpanWorld has smaller error increases along all three paths; its smallest increase occurs after 15+35 days, with an interval spanning zero. Table C3 reports paired between-model differences in degradation, all negative. Table C4 reports between-model differences

at day 100. The direct-prediction difference is positive with an interval crossing zero, whereas all three continued-prediction differences are negative. These results distinguish direct accuracy from post-rollout prediction.

Table C1: Absolute errors and between-model differences on the common suffix. Days 55–100, N=2,837. Differences are SpanWorld minus the direct-endpoint control; negative values favor SpanWorld. Brackets give 95% intervals.

Starting state	Control MSE [95% CI]	SpanWorld MSE [95% CI]	ΔMSE [95% CI]
Initial-state direct	0.041480 [0.035780, 0.047559]	**0.041285** [0.035709, 0.047239]	-0.000195 [-0.000382, -0.000015]
50	0.043713 [0.038079, 0.049802]	**0.041688** [0.036009, 0.047699]	-0.002025 [-0.002680, -0.001300]
25+25	0.046488 [0.040882, 0.052690]	**0.041536** [0.035868, 0.047549]	-0.004952 [-0.005948, -0.003868]
15+35	0.046550 [0.040938, 0.052730]	**0.041448** [0.035763, 0.047426]	-0.005101 [-0.006107, -0.004009]

* Within each row, the lower model MSE is bolded.

Table C2: Change in subsequent error relative to each model's direct prediction. Same window, eligibility set, and spatial resampling as Table 1; positive values indicate increased error.

Model	Route to day 50	ΔMSE vs own direct	95% CI
Direct-endpoint control	50	0.002234	[0.001448, 0.002960]
Direct-endpoint control	25+25	0.005008	[0.003895, 0.006040]
Direct-endpoint control	15+35	0.005070	[0.003952, 0.006097]
SpanWorld	50	**0.000403**	[0.000231, 0.000582]
SpanWorld	25+25	**0.000251**	[0.000078, 0.000440]
SpanWorld	15+35	**0.000163**	[-0.000010, 0.000345]

* For each path, the lower increase in MSE is bolded.

Table C3: Between-model differences in degradation. Differences equal (SpanWorld continued-own direct)-(control continued-own direct), with N=2,837 and the same paired resampling.

Route to day 50	Difference in degradation	95% CI
50	-0.001831	[-0.002500, -0.001076]
25+25	-0.004757	[-0.005780, -0.003649]
15+35	-0.004906	[-0.005929, -0.003788]

Table C4: Between-model differences on the separately eligible day-100 set. Differences are SpanWorld minus the direct-endpoint control; $N = 1,670$, with 80 geographic groups.

Starting state	Day-100 ΔMSE	95% CI
Initial state	0.000126	[-0.000065, 0.000317]
50-day state	-0.001360	[-0.002508, -0.000191]
25+25-day state	-0.004292	[-0.005976, -0.002617]
15+35-day state	-0.004723	[-0.006394, -0.003044]

D Evaluation on Unseen Temporal Partitions

D.1 Partitions and Evaluation Definitions

We evaluate nine temporal partitions absent from the training candidates in Appendix A.3 using the frozen SpanWorld checkpoint on the 2,856-sample IID manifest; all three endpoint dates receive supervision of direct predictions. Thus, unseen refers specifically to the temporal partition. This experiment compares direct and partitioned predictions at the same endpoint, without subsequent queries from that endpoint. Pooled MSE divides the total squared error by the total number of jointly valid pixels. There are 1,615, 1,568, and 1,617 eligible samples at days 50, 75, and 100, respectively. This estimand differs from the sample-averaged subsequent-window MSE in Appendix C.

D.2 Full Endpoint Results

Table D1 lists all nine partitions. The partitioned-to-direct MSE ratios range from 0.992329 to 1.023419, corresponding to changes of approximately -0.77% to +2.34%. Errors both increase and decrease. Endpoint accuracy remains close to direct prediction over the evaluated partitions.

Table D1: Endpoint results for nine training-unseen partitions. Ratios divide partitioned MSE by direct MSE at the same date; segment lengths are in days. Figure 4(a) uses $100\times$(ratio-1), computed from unrounded source values.

Day	Segments (days)	N	Direct MSE	Partitioned MSE	Ratio	Change (%)
50	15+35	1615	**0.027289**	0.027750	1.016875	1.687
50	30+20	1615	**0.027289**	0.027772	1.017694	1.769
50	10+15+25	1615	**0.027289**	0.027929	1.023419	2.342
75	20+55	1568	**0.035779**	0.036384	1.016923	1.692
75	15+25+35	1568	**0.035779**	0.036561	1.021870	2.187
100	40+60	1617	0.037691	**0.037401**	0.992329	-0.767
100	10+90	1617	**0.037691**	0.037893	1.005365	0.536
100	10+30+60	1617	0.037691	**0.037591**	0.997369	-0.263
100	5+20+30+45	1617	0.037691	**0.037478**	0.994352	-0.565

* Within each row, the lower direct or partitioned MSE is bolded.

E Contributions of Model-Generated States

E.1 Intervention and Fixed Conditions

Using the fixed SpanWorld checkpoint, we form the intermediate state with a single 50-day transition and replace it with a matched donor's state at the same time. The receiver's historical baseline, geography, subsequent weather, and target dates remain fixed. Donors share the split and observation-start year/month, with self-matches excluded. Four summaries use the first ten observations: masked NDVI frame mean, frame standard deviation, last valid-frame mean, and valid-frame fraction. Features are standardized by training-calibration medians and IQRs, replacing zero or nonfinite IQRs with one. Distance combines Euclidean distance in this four-dimensional space with 0.5 times the Hellinger distance between normalized 256-bin land-cover histograms. Receivers are processed in relative-path order; nearest-neighbor ties are resolved by donor path, with at most one use per donor. Samples with missing features or no match are excluded from both conditions. Matching uses only the historical summaries, land-cover composition, eligibility criteria, and deterministic assignment rules described above.

We evaluate days 55–100 using each receiver's observation mask, shared between original-state and donor-state predictions. Each manifest entry with its context and forecast window defines a receiver. Positive donor-minus-original MSE differences indicate harm from replacement. Table E1 separates the full subsequent window from the day-100 eligibility set. Sample-equal weighting is primary; group-equal and pixel-pooled results assess weighting sensitivity. Each estimate has a corresponding

paired geographic-bootstrap 95% interval from 2,000 resamples with seed 20260910, conditional on the fixed matching and trained model.

E.2 Full Replacement Results

Primary differences over days 55–100 are positive on all four splits, with intervals above zero; the group-equal interval for day 100 on OOD-ST spans zero. The subsequent-window results show that the original sample's generated state contributes to continued prediction when its historical baseline, geography, and subsequent weather are retained.

Table E1: State-replacement MSE differences and 95% intervals. Each cell is an estimate [lower, upper]; the three weighting schemes are distinct.

Days	Split	N	Groups	ΔMSE	95% CI
			Sample-equal		
55–100	IID	2066	80	0.003192	[0.002537, 0.003848]
55–100	OOD-T	1605	80	0.002192	[0.001660, 0.002708]
55–100	OOD-S	6965	878	0.003345	[0.002975, 0.003719]
55–100	OOD-ST	5437	878	0.001558	[0.001247, 0.001867]
100	IID	1214	80	0.004109	[0.002808, 0.005303]
100	OOD-T	884	80	0.003140	[0.002143, 0.004180]
100	OOD-S	3549	852	0.003431	[0.002734, 0.004111]
100	OOD-ST	2731	854	0.001004	[0.000364, 0.001668]
			Group-equal		
55–100	IID	2066	80	0.003168	[0.002420, 0.003897]
55–100	OOD-T	1605	80	0.002246	[0.001697, 0.002795]
55–100	OOD-S	6965	878	0.003419	[0.003054, 0.003793]
55–100	OOD-ST	5437	878	0.001558	[0.001237, 0.001878]
100	IID	1214	80	0.003677	[0.002313, 0.004965]
100	OOD-T	884	80	0.003228	[0.001965, 0.004465]
100	OOD-S	3549	852	0.002767	[0.001892, 0.003610]
100	OOD-ST	2731	854	0.000533	[-0.000222, 0.001314]
			Pixel-pooled		
55–100	IID	2066	80	0.003399	[0.002657, 0.004120]
55–100	OOD-T	1605	80	0.002368	[0.001784, 0.002967]
55–100	OOD-S	6965	878	0.002617	[0.002200, 0.003028]
55–100	OOD-ST	5437	878	0.001284	[0.000941, 0.001633]
100	IID	1214	80	0.004336	[0.003215, 0.005373]
100	OOD-T	884	80	0.002431	[0.001708, 0.003112]
100	OOD-S	3549	852	0.002982	[0.002344, 0.003539]
100	OOD-ST	2731	854	0.000794	[0.000186, 0.001406]

Additional frozen-model interventions set the state-correction coefficient to zero ($\widehat{Y}_t = B_t$) or replace the shared transition with the identity map (retaining the correction from S_0). Both preserve historical information, without retraining. Table E2 reports standard forecasting metrics over the full 100-day forecast window. Main-text Table 2 presents its RMSE results.

F Weather Inputs and Forecast Responses

F.1 Weather Conditions and Pairing Protocol

We evaluate the same SpanWorld checkpoint on a frozen OOD-T hot-dry subset. Selection uses the means of daily temperature and precipitation over the next 100 days. Heat is a standardized temperature anomaly; dryness is the sign-reversed standardized precipitation anomaly. Training climatology is stratified by forecast-start month and 3° latitude–longitude cells, falling back to month×6° latitude bands and then month when fewer than 30 records or invalid variance prevent calibration. The

Table E2: Removing the state correction or using an identity transition. Full 100-day forecast window; official R^2_{LC} and $\mathrm{RMSE}_{\mathrm{LC}}$.

Split	Full R^2	No correction R^2	Identity transition R^2	Full RMSE	No correction RMSE	Identity transition RMSE
IID	**0.5220**	<u>0.4883</u>	0.4877	**0.1555**	<u>0.1708</u>	0.2150
OOD-T	**0.5726**	<u>0.5555</u>	0.5549	**0.1509**	<u>0.1627</u>	0.2150
OOD-S	**0.4949**	<u>0.4614</u>	0.4605	**0.1621**	<u>0.1755</u>	0.2196
OOD-ST	**0.5408**	<u>0.5237</u>	0.5232	**0.1542**	<u>0.1650</u>	0.2164

* Within each split and metric, best results are bolded and second-best results are underlined.

predefined 80th-percentile tier requires heat$\geq$0.845641, dryness$\geq$0.820556, and jointly valid future temperature and precipitation on at least 80% of days, yielding 84 samples. This coverage criterion applies to the joint validity of future temperature and precipitation records.

Normal-weather donors have absolute heat and dryness anomalies $\leq$0.75 and share the receiver's meteorological season (DJF/MAM/JJA/SON). Eight matching features are day of year, latitude, longitude, vegetation fraction, historical cloud fraction, historical observation-validity fraction, historical mean NDVI, and future weather-validity fraction. Features are standardized by means and standard deviations of the combined extreme/normal candidate pool; root-mean-square standardized distance must be $\leq$1.5. Deterministic sample-ID ordering and a maximum of five uses per donor yield 84 pairs with 45 unique donors. Matching uses neither future NDVI nor prediction errors.

We retain each receiver's history, geography, and model weights while replacing all 20 future weather steps with the donor's ordered sequence or zeros in normalized weather space. The latter gives normalized-mean weather conditions. Losses cover valid vegetation pixels over the full 100-day window and are averaged equally across samples. Differences are replacement-minus-actual MSE. We use 10,000 bootstrap resamples, separately resampling the 84 pairs, 31 geographic groups, or 45 reused-donor groups; Figure 4(c) uses geographic-group intervals.

F.2 Prediction Responses and Factual Errors

Both replacements increase hot-dry forecasting error, with a larger increase under mean weather (Table F1). Mean absolute changes between predictions are 0.034663 for donor and 0.079716 for mean weather; these measure mean absolute differences between predictions before and after weather replacement. The hot-dry-minus-normal loss interaction under donor replacement has an interval crossing zero, with no clear evidence of a larger error increase on the hot-dry subset. All errors are evaluated against observations under the actual weather.

G Qualitative Forecast Cases

G.1 Case Selection and Display Protocol

Main-text Figure 3 uses IID sample P42 (manifest index 2055, `MJJ19/minicube_23_30STJ_39.19_-5.73.nc`) at day 100 (2019-08-05). Each model first forms its day-50 state through 15+35 days of rollout and then predicts day 100; direct forecasts query the same date from the initial state, without new imagery.

The candidate pool retained 1,340 of the 2,837 common-window samples with suffix coverage ≥ 0.40. It included cases at five fixed quantiles (10%, 25%, 50%, 75%, 90%) of control-minus-SpanWorld suffix MSE, plus the preselected P42. Figure 3 was selected from 41 case–target-date combinations with coverage ≥ 0.40 across these six cases, based on the between-model difference in single-date degradation. P42 illustrates a marked improvement within this restricted pool.

All models and routes share observations and validity masks. NDVI panels share one color scale, and absolute-error panels share another. Gray denotes invalid or unscored pixels; error maps show absolute differences between predictions and observations. NDVI and absolute-error display ranges

Table F1: Loss differences under weather replacement and cohort interactions. The first two rows report replacement-minus-actual MSE on the hot-dry subset; the last two report hot-dry-minus-matched-normal differences in this effect.

Resampling unit	ΔMSE	95% CI
	Donor-actual	
Paired	0.002053	[0.000793, 0.003352]
Geographic	0.002053	[0.000761, 0.003452]
Donor-group	0.002053	[0.000715, 0.003361]
	Mean-actual	
Paired	0.010140	[0.006782, 0.013770]
Geographic	0.010140	[0.004839, 0.015558]
Donor-group	0.010140	[0.004760, 0.015883]
	Donor effect: hot-dry-normal	
Paired	-0.000302	[-0.002321, 0.001759]
Geographic	-0.000302	[-0.002841, 0.002582]
Donor-group	-0.000302	[-0.003153, 0.002443]
	Mean effect: hot-dry-normal	
Paired	0.007838	[0.004395, 0.011532]
Geographic	0.007838	[0.003225, 0.011961]
Donor-group	0.007838	[0.002767, 0.013311]

are [-0.6, 0.9] and [0, 0.45]. MSE is computed from predictions before display clipping. Figure 3 scores 16,147 of 16,384 pixels. Direct/continued frame MSEs are 0.005882/0.004518 for SpanWorld and 0.005390/0.042048 for the control. Aggregate results appear in Table 1 and Appendix C.

G.2 Additional Cases

Figure G1 uses P42 at days 25 and 75, fixed before rendering rather than selected by frame error. Figure G2 uses the tenth-percentile case Q10 (manifest index 2132, `MJJ19/minicube_91_31UGS_50.56_5.89.nc`). Day 70 (2019-07-05) is selected solely for its maximum valid-pixel coverage (0.9682) within the subsequent window. Both figures follow the display and scoring rules in G.1.

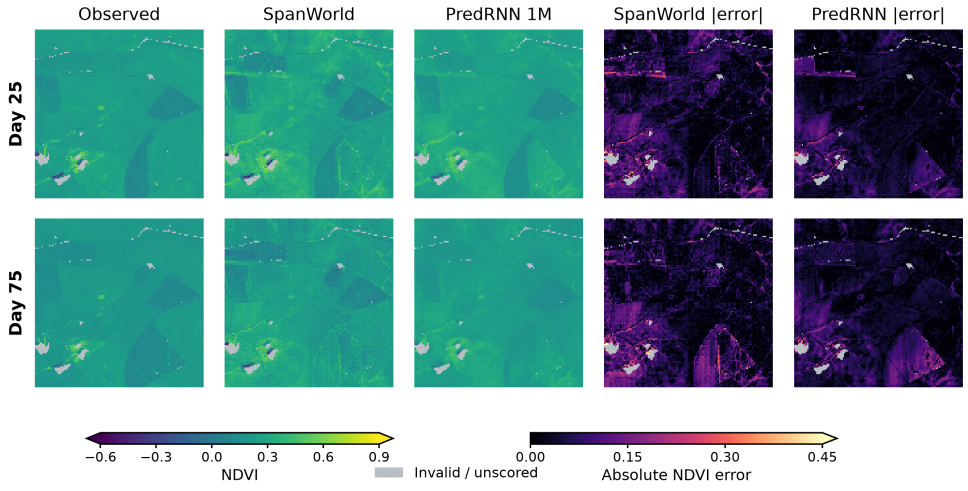


Figure G1: Additional standard forecasts. Observations, SpanWorld and official PredRNN 1M predictions, and absolute errors at days 25/75 for P42. Frame MSEs for SpanWorld/PredRNN are 0.005089/0.002808 and 0.005728/0.003597; PredRNN has lower error in this example.

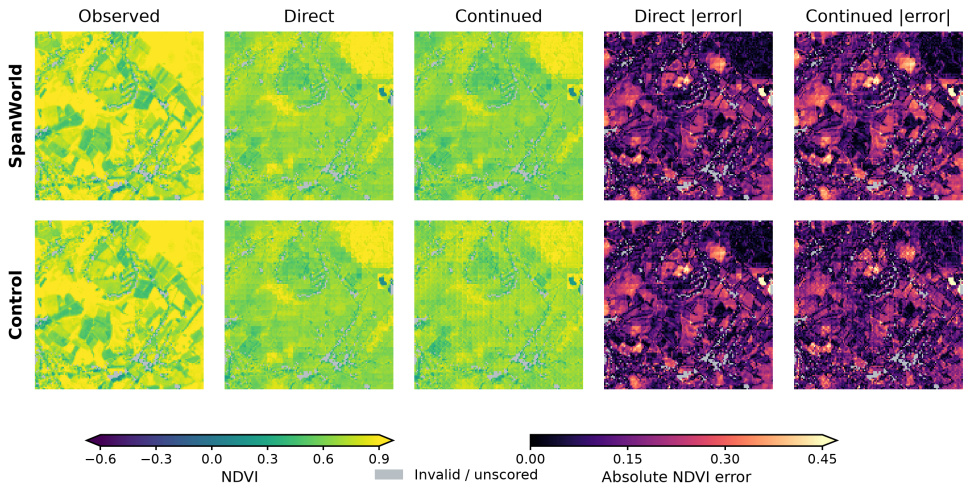


Figure G2: Degradation in continued prediction. Q10 at day 70 after a 15+35-day rollout to day 50. SpanWorld is shown above the direct-endpoint control; Direct starts from the initial state and Continued from the generated state. Direct/continued MSEs are 0.024007/0.030284 and 0.025428/0.024308, illustrating a case of degradation despite the aggregate improvement.

H Resources, Checkpoints, and Scope

Table H1 summarizes resources for the formal training stages that produced the checkpoints used in the core comparison. SpanWorld and its same-architecture direct-endpoint control start from the same trained parent weights; the table lists devices, update counts, and measured compute.

The evidence concerns given weather and a finite forecast window, with variable spans in integer multiples of five-day steps. No new remote-sensing observation or online assimilation enters during prediction.

Table H1: Current formal stages mapped to the core-result checkpoints. Allocated GPU-hours = 4 × elapsed wall-hours; earlier shared-parent training is excluded.

Resource	SpanWorld	Direct-endpoint control
Parameters	7,180,896	7,180,896
Devices × batch/device × accumulation	4 × 8 × 2	4 × 8 × 2
Global batch	64	64
Updates	14,880	14,880
Wall-hours	5.901	5.665
Allocated GPU-hours	23.606	22.661