Supplementary Material: Multi-modal learning for geospatial vegetation forecasting

Vitus Benson[1,2,3,*] Claire Robin[1,2] Christian Requena-Mesa[1,2] Lazaro Alonso[1]
Nuno Carvalhais[1,2] José Cortés[1] Zhihan Gao[4] Nora Linscheid[1] Mélanie Weynants[1]
Markus Reichstein[1,2]
[1] Max-Planck-Institute for Biogeochemistry [2] ELLIS Unit Jena [3] ETH Zürich
[4] Hong Kong University of Science and Technology [*] vbenson@bgc-jena.mpg.de

A. Model details

A.1. Cloud masking

Baselines (Table 1) The baselines reported in table 1 are taken from CloudSEN12 [2]. Sen2Cor [14] is the processing software from ESA used to produce the Scene Classification Layer (SCL) mask, which was also introduced in EarthNet2021 [18]. FMask [17] is a processing software originally designed for NASA Landsat imagery, but now repurposed to also work with Sentinel 2 imagery. It requires L1C top-of-atmosphere reflectance from all bands to be produced (EarthNet2021 only containes L2A bottom-of-atmosphere reflectance from four bands). KappaMask [6] is a cloud mask based on deep learning, in table 1 we reported scores from the L2A version, which uses all 13 L2A bands as input.

UNet Mobilenetv2 (Table 1) Our UNet with Mobilenetv2 encoder [20] was trained in two variants, one with RGB and near-infrared bands of L2A imagery (i.e. works with EarthNet2021) and one with all 13 bands of L2A imagery. We adopted the exact same implementation that was benchmarked in the CloudSEN12 paper [2], with the only difference being that in the paper, L1C imagery was used (which is often not useful in practical use-cases). In detail, this means we trained the UNet with Mobilenetv2 encoder using the Segmentation Models PyTorch Python library[1]. We used a batch size of 32, random horizontal and vertical flipping, random 90 degree rotations, random mirroring, unweighted cross entropy loss, early stopping with a patience of 10 epochs, AdamW optimizer, learning rate of $1e^{-3}$, and a learning rate schedule reducing the learning rate by a factor of 10 if validation loss did not decrease for 4 epochs.

A.2. Vegetation modeling

Implementation details We build all of our ConvNets with a PatchMerge-style architecture similar to the one in Earthformer [7]. For SimVP and PredRNN, such encoders and decoders are more powerful, but also slightly more parameter-intensive, than the variants used in the original papers. We use GroupNorm [27] and LeakyReLU activation [28] for the ConvNets, and and ConvLSTMs. For the Contextformer, we use LayerNorm [3] and GELU activation [8]. For ConvNets, skip connections preserve high-fidelity content between encoders and decoders. Our framework is implemented in PyTorch, and models are trained on Nvidia A40 and A100 GPUs. We use the AdamW [13] optimizer and tune the learning rate and a few hyperparameters per model. More implementation details can be found in the supplementary materials.

Contextformer (Table 2,3,4,5 Figure 3,4,5) Our Contextformer is a combination of a spatial vision encoder: Pyramid Vision Transformer (PVT) v2 B0 [23, 24] with pre-trained ImageNet1k weights from the PyTorch Image Models library[2] and a temporal transformer encoder. We use a patch size of 4×4 px. We use an embedding size of 256 and the temporal transformer has three self-attention layers with 8 heads, each followed by an MLP with 1024 hidden channels. We use LayerNorm [3] for normalization and GELU [8] as non-linear activation function. The model is trained with masked token modeling, randomly ($p = 0.5$) flipping between inference mask (future token masked) and random dropout mask (70% of image patches masked, except for the first 3 time steps). We train for 100 epochs with a batch size of 32, a learning rate of $4e^{-5}$ and with AdamW optimizer on 2 NVIDIA A100 GPUs. We train three models from the random seeds 42, 97 and 27.

[1] https://segmentation-models-pytorch.readthedocs.io/en/latest/

[2] https://github.com/rwightman/pytorch-image-models

Local timeseries models (Table 2) We train the local timeseries models (table 2) at each pixel. For a given pixel we extract the full timeseries of NDVI and weather variables at 5-daily resolution. All variables are linearly gap-filled and weather is aggregated with min, mean, max, and std to 5-daily. The whole timeseries before each target period is used to train a timeseries model, for the target period the model only receives weather. The Kalman Filter runs with default parameters from darts [9]. The LightGBM model gets lagged variables from the last 10 time steps and predicts a full 20 time step chunk at once. For Prophet we again use default parameters.

EarthNet models (Table 2) For running the leading models from EarthNet2021 we utilize the code from the respective github repositories: ConvLSTM [5][3], SGED-ConvLSTM [11][4] and Earthformer [7] [5]. We derive the NDVI from the predicted satellite bands red and near-infrared:

$$NDVI = \frac{NIR - Red}{NIR + Red + 1e^{-8}} \tag{1}$$

ConvLSTM (Table 2,3, Figure 4,5) Our ConvLSTM contains four ConvLSTM-cells [21] in total, two for processing context frames and two for processing target frames. Each has convolution kernels with bias, hidden dimension of 64 and kernel size of 3. We train for 100 epochs with a batch size of 32, a learning rate of $4e^{-5}$ and with AdamW optimizer. We train three models from the random seeds 42, 97 and 27.

PredRNN (Table 2,3, Figure 4) Our PredRNN contains two ST-ConvLSTM-cells [25] Each has convolution kernels with bias, hidden dimension of 64 and kernel size of 3 and residual connections. We use a PatchMerge encoder decoder with GroupNorm (16 groups), convolutions with kernel size of 3 and hidden dimension of 64, LeakyReLU activation and downsampling rate of 4x. We train for 100 epochs with a batch size of 32, a learning rate of $3e^{-4}$ and with AdamW optimizer. We use a spatio-temporal memory decoupling loss term with weight 0.1 and reverse exponential scheduling of true vs. predicted images (as in the PredRNN journal version [26]). We train three models from the random seeds 42, 97 and 27.

SimVP (Table 2,3, Figure 4) Our SimVP has a PatchMerge encoder decoder with GroupNorm (16 groups), convolutions with kernel size of 3 and hidden dimension of 64, LeakyReLU activation and downsampling rate of 4x. The encoder processes all 10 context time steps at once (stacked along the channel dimension). The decoder processes 1 target time step at a time. The gated spatio-temporal attention processor [22] translates between both in the latent space, we use two layers and 64 hidden channels. We train for 100 epochs with a batch size of 64, a learning rate of $6e^{-4}$ and with AdamW optimizer. We train three models from the random seeds 42, 97 and 27.

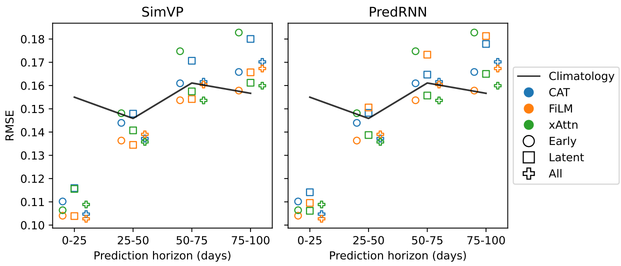

Figure 5. Model performance (RMSE) when using different ways of weather conditioning over varying prediction horizons.

Earthformer (Table 2) Our Earthformer is a transformer combined with an initial PatchMerge encoder (and a final decoder) to reduce the dimensionality. The encoder and decoder use LeakyReLU activation, hidden size of 64 and 256 and downsample 2x. In between, the transformer processor has a UNet-type architecture, with cross-attention to merge context frame information with target frame embeddings. GeLU activation and LayerNorm, axial self-attention, 0.1 dropout and 4 attention heads are used. Weather information is regridded to match the spatial resolution of satellite imagery and used as input during context and target period. We train for 100 epochs with a batch size of 32, a maximum learning rate of $1e^{-4}$, linear learning rate warm up, cosine learning rate shedule and with AdamW optimizer.

1x1 LSTM (Table 4) Our 1x1 LSTM is implemented as a ConvLSTM with kernel size of 1. We train for 100 epochs with a batch size of 32, a learning rate of $4e^{-5}$ and with AdamW optimizer.

Next-frame UNet (Table 4) Our next-frame UNet has a depth of 5, latent weather conditioning with FiLM, a hidden size 128, kernel size 3, LeakyReLU activation, GroupNorm (16 groups), PatchMerge downsampling and nearest upsampling. We train for 100 epochs with a batch size of 64, a learning rate of $6e^{-4}$ and with AdamW optimizer.

Next-cuboid UNet (Table 4) Our next-cuboid UNet has a depth of 5, latent weather conditioning with FiLM, a hid-

[3] https://github.com/dcodrut/weather2land

[4] https://github.com/rudolfwilliam/satellite_image_forecasting

[5] https://github.com/amazon-science/earth-forecasting-transformer/tree/main/scripts/cuboid_transformer/earthnet_w_meso

	OOD-s		OOD-st	
Model	$R^2 \uparrow$	RMSE $\downarrow$	$R^2 \uparrow$	RMSE $\downarrow$
Climatology	0.50	0.15	0.56	0.19
ConvLSTM	0.47	0.17	0.52	0.16
Earthformer	0.47	0.15	0.47	0.16
PredRNN	0.54	0.15	0.58	0.15
SimVP	0.50	0.15	0.54	0.15
Contextformer	0.54	0.15	0.58	0.14

Table 6. Same as table 5, but extended. Model skill at spatial (OOD-s) and spatio-temporal (OOD-st) extrapolation.

den size 256, kernel size 3, LeakyReLU activation, GroupNorm (16 groups), PatchMerge downsampling and nearest upsampling. We train for 100 epochs with a batch size of 64, a learning rate of $6e^{-4}$ and with AdamW optimizer.

B. Weather ablations

B.1. Methods

Most of our baseline approaches have been originally proposed to handle only past covariates. Here, we condition forecasts on future weather. A-priori it is not known how to best achieve this weather conditioning. For PredRNN and SimVP, we compare three approaches, each fused at three different locations. The approaches operate pixelwise, taking features $x_{in} \in \mathbb{R}^d$ and conditioning input $c_i \in \mathbb{R}^n$ for weather variable i. The conditioning layers $g(\cdot, \cdot; \phi)$ with parameters ϕ then operate as

$$x_{out} = g(x_{in}, c; \phi) \in \mathbb{R}^d \tag{2}$$

We parameterize g with neural networks.

CAT First concatenates x_{in} and a flattened c along the channel dimension, and then performs a linear projection to obtain x_{out} of same dimensionality as x_{in}. In practice we implement this with a 1x1 Conv layer.

FiLM Feature-wise linear modulation [16] generalizes the concatenation layer before. It produces x_{out} with linear modulation:

$$x_{out} = x_{in} + \sigma(\gamma(c; \phi_\gamma) \odot N(f(x_{in}; \phi_f)) + \beta(c; \phi_\beta)) \tag{3}$$

Here, f is a linear layer, γ and β are MLPs, N is a normalization layer and σ is a pointwise non-linear activation function.

xAttn Cross-attention is an operation commonly found in the Transformers architecture. In recent works on image generation with diffusion models it is used to condition the generative process on a text embedding [19]. Inspired from this, we propose a pixelwise conditioning layer based on multi-head cross-attention. The input x_{in} is treated as a single token query Q. Each weather variable c_i is treated as individual tokens, from which we derive keys K and values V. The result is then just regular multi-head attention MHA in a residual block:

$$\begin{aligned} x_{out} &= x_{in} && (4) \\ &\quad + f(N(MHA(Q(x_{in}; \phi_Q), K(c; \phi_K), V(c; \phi_V))); \phi_f) && (5) \end{aligned}$$

Here, f is either a linear projection or a MLP and N is a normalization layer.

Each of the three approaches we apply at three locations throughout the network:

Early fusion Just fusing all data modalities before passing it to a model. This Early CAT has been previously used for weather conditioning in satellite imagery forecasting

Latent fusion In the encode-process-decode framework, encoders are meant to capture spatial, and not temporal, relationships. Hence, latent fusion conditions the encoded spatial inputs twice: right after leaving the encoder and before entering the decoder.

All (fusion everywhere) In addition, we compare against conditioning at every stage of the encoders, processors and decoders. All CAT has been applied to condition stochastic video predictions on random latent codes [12].

B.2. Results

Fig. 5 summarizes the findings by looking at the RMSE over the prediction horizon. For the first 50 days, most models are better than the climatology, afterwards, most are worse. If using early fusion, FiLM is the best conditioning method. For latent fusion and fusion everywhere (all), xAttn is a consistent choice, but FiLM may sometimes be better (and sometimes a lot worse). CAT in general should be avoided, which is consistent with the theoretical observation, that CAT is a special case of FiLM.

For SimVP, the best weather guiding method is latent fusion with FiLM. For PredRNN, the best method is early fusion with FiLM. This is likely due to the difference in treatment of the temporal axis. For SimVP, early fusion would merge all time steps, hence, latent fusion is a better choice. For PredRNN on the other hand, early fusion handles only a single timestep.

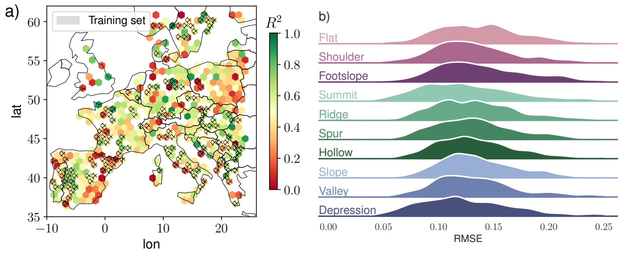

Figure 6. Panel a) shows a map of R^2 on OOD-t and OOD-st test sets and panel b) shows probability densities of RMSE per geomorphon. Both for Contextformer.

C. Contextformer Strengths and Limitations continued

We show spatial extrapolation skills for more models in table 6.

Reassured by spatial extrapolation capabilities, we present a map of R^2 for the Contextformer in fig. 6a. Cropland regions on the Iberian peninsula and in northern France, as well as forests in the Balkans are regions with great applicability of the model. For the former two, this may be explained by many training samples in those regions, for the last, it cannot. Grasslands and forests in Poland and highly heterogenous regions (mountains, near cities, near coasts) are more challenging for the model.

Geomorphons capture local terrain features, derived from first and second spatial derivatives of elevation. Fig. 6b shows densities of RMSE of the ConvLSTM for different geomorphons from the Geomorpho90m map [1]. Generally, the model performs well across all classes. Summits and Depressions, two rather extreme types, seem to be slightly easier to predict. Homogeneous terrain (red: flat, shoulder, footslope) has a larger tail towards high error. This may be as those regions are typically where there is a lot of anthropogenic activity, possibly leading to dynamics less covered by the predictors (harvest, clear-cut, etc.).

The OOD-t test set includes minicubes from four 3-month periods over two years. Fig. 7 shows Contextformer's model skill. Yearly variations are significant. Growing season prediction was better in 2022 until September, then it switched, and 2021 performed better. First half of the season is usually better predicted than the second half, likely due to anthropogenic influences (harvest, mowing, cutting, and forest fires). These events are challenging to predict from weather covariates and may be interpreted as random noise.

D. Performance per landcover type

Fig. 8 shows the model performance per landcover type.

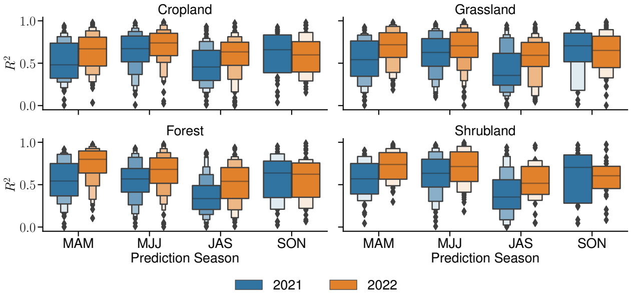

Figure 7. Contextformer model skill for different seasons and landcover on the OOD-t test set.

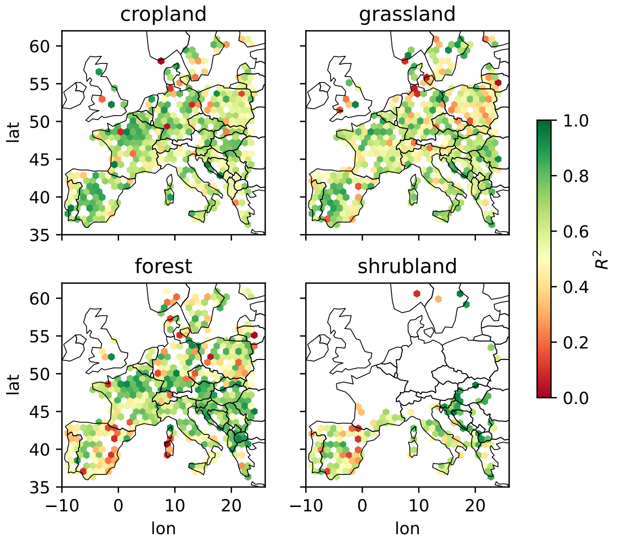

Figure 8. Model performance per landcover. Maps represent R^2 on OOD-t and OOD-st test sets of PredRNN.

E. Robustness of Outperformance Score

The choice of thresholds in the outperformance score (the percentage of samples where a model outperforms the climatology baseline by at least the threshold on at least 3 out of 4 metrics) is a heuristic. To assess its robustness, we reevaluated five of our models over a wide range of possible threshold values. Fig. 9 shows a consistency of the ranking, in particular our Contextformer outperforms all other models in all settings.

F. Inference speed

Computing inference speed is highly platform and batch size dependent. To make it somewhat fair, we compare models by running 1024 samples on an A40 GPU (48GB), with the largest batch size (bs) fitting in memory, we perform 10 repetitions and report the mean and std. dev.: Contextformer 29.3s±0.4 (bs 72), SimVP 6.7s±0.8 (bs 96), PredRNN 16.2s±0.2 (bs 512), ConvLSTM 37.1s±1.8 (bs 256). For comparison predicting a single sample with one of the

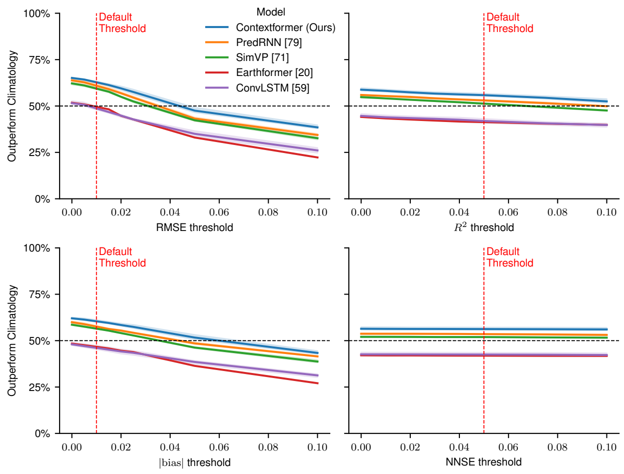

Figure 9. Outperformance score robustness (shown: marginals).

local time series models takes >1h on a single CPU.

G. Downstream task: carbon monitoring

Carbon monitoring is of great importance for climate change mitigation, especially in relation to nature-based solutions. The gross primary productivity (GPP) represents the amount of carbon that is taken up by plants through photosynthesis and subsequently stored. It is not directly observable. At a few hundred research stations around the world with eddy covariance measurement technology, it can be indirectly measured. For carbon monitoring, it would be beneficial to measure this quantity everywhere on the globe. It has been shown [15] that Sentinel 2 NDVI is correlated to GPP measured with eddy covariance. We build on this correlation to show how our models could potentially be leveraged to give near real-time estimates of GPP and to study weather scenarios.

Fig. 10 compares modeled with observed GPP at the Fluxnet site Grillenburg (identifier DE-Gri) in eastern Germany distributed by ICOS [10]. First, we fit a linear model between observed NDVI and GPP for the years 2017-2019. Here, interpolated grassland NDVI pixels (fig. 10b, inside red boundaries) are used. Next, we perform an out-of-sample analysis and find an $R^2 = 0.53$ for 2020-01 to 2021-04 (fig. 10a, blue line). Finally, we forecast GPP with our PredRNN model from May to July 2021(fig. 10a, orange line). The resulting forecast has decent quality at short prediction horizons, but low skill after 75 days (fig. 10c). These results show a way to leverage models from this paper for near real-time carbon monitoring. However, for application at scale, it is likely beneficial to use a more powerful GPP model (e.g. random forest [15] or light-use efficiency [4]), fitted across many Fluxnet sites.

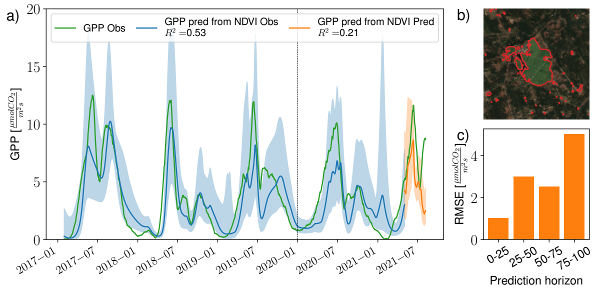

Figure 10. Panel a) shows timeseries of observed (green) and modeled GPP (blue from NDVI observations, orange from NDVI prediction). Panel b) shows a satellite image of the Grillenburg Fluxnet site with grassland boundaries in red. Panel c) shows the RMSE over prediction horizons.

References

[1] Giuseppe Amatulli, Daniel McInerney, Tushar Sethi, Peter Strobl, and Sami Domisch. Geomorpho90m, empirical evaluation and accuracy assessment of global high-resolution geomorphometric layers. *Scientific Data*, 7(1):162, 2020. 4

[2] Cesar Aybar, Luis Ysuhuaylas, Jhomira Loja, Karen Gonzales, Fernando Herrera, Lesly Bautista, Roy Yali, Angie Flores, Lissette Diaz, Nicole Cuenca, Wendy Espinoza, Fernando Prudencio, Valeria Llactayo, David Montero, Martin Sudmanns, Dirk Tiede, Gonzalo Mateo-García, and Luis Gómez-Chova. CloudSEN12, a global dataset for semantic understanding of cloud and cloud shadow in Sentinel-2. *Scientific Data*, 9(1):782, 2022. 1

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv*, 1607.06450, 2016. 1

[4] Shanning Bao, Andreas Ibrom, Georg Wohlfahrt, Sujan Koirala, Mirco Migliavacca, Qian Zhang, and Nuno Carvalhais. Narrow but robust advantages in two-big-leaf light use efficiency models over big-leaf light use efficiency models at ecosystem level. *Agricultural and Forest Meteorology*, 326: 109185, 2022. 5

[5] Codruț-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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1362–1371, 2022. 2

[6] Marharyta Domnich, Indrek Sünter, Heido Trofimov, Olga Wold, Fariha Harun, Anton Kostiukhin, Mihkel Järveoja, Mihkel Veske, Tanel Tamm, Kaupo Voormansik, Aire Olesk, Valentina Boccia, Nicolas Longepe, and Enrico Giuseppe Cadau. KappaMask: AI-Based Cloudmask Processor for Sentinel-2. *Remote Sensing*, 13(20):4100, 2021. 1

[7] Zhihan Gao, Xingjian Shi, Hao Wang, Yi Zhu, Bernie Wang, Mu Li, and Dit-Yan Yeung. Earthformer: Exploring Space-Time Transformers for Earth System Forecasting. In *Advances in Neural Information Processing Systems*, 2022. 1, 2

[8] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv*, 1606.08415, 2023. 1

[9] Julien Herzen, Francesco Lässig, Samuele Giuliano Piazzetta, Thomas Neuer, Léo Tafti, Guillaume Raille, Tomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki, Nicolas Huguenin, Maxime Dumonal, Jan Kościsz, Dennis Bader, Frédérick Gusset, Mounir Benheddi, Camila Williamson, Michal Kosinski, Matej Petrik, and Gaël Grosch. Darts: User-Friendly Modern Machine Learning for Time Series. *Journal of Machine Learning Research*, 23 (124):1–6, 2022. 2

[10] ICOS RI. Ecosystem final quality (l2) product in etc-archive format - release 2022-1. station de-gri. 2022. 5

[11] Klaus-Rudolf Kladny, Marco Milanta, Oto Mraz, Koen Hufkens, and Benjamin D Stocker. Enhanced prediction of vegetation responses to extreme drought using deep learning and earth observation data. *Ecological Informatics*, 80: 102474, 2024. 2

[12] Alex X. Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic Adversarial Video Prediction. *arxiv*, 1804.01523, 2018. 3

[13] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2022. 1

[14] Jérôme Louis, Vincent Debaecker, Bringfried Pflug, Magdalena Main-Knorn, Jakub Bieniarz, Uwe Mueller-Wilm, Enrico Cadau, and Ferran Gascon. SENTINEL-2 SEN2COR: L2A Processor for Users. In *Proceedings Living Planet Symposium 2016*, pages 1–8, Prague, Czech Republic, 2016. Spacebooks Online. 1

[15] Daniel E. Pabon-Moreno, Mirco Migliavacca, Markus Reichstein, and Miguel D. Mahecha. On the potential of Sentinel-2 for estimating Gross Primary Production. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–1, 2022. 5

[16] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual Reasoning with a General Conditioning Layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. 3

[17] Shi Qiu, Zhe Zhu, and Binbin He. Fmask 4.0: Improved cloud and cloud shadow detection in Landsats 4–8 and Sentinel-2 imagery. *Remote Sensing of Environment*, 231: 111205, 2019. 1

[18] 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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1132–1142, 2021. 1

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 3

[20] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 1

[21] 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*. Curran Associates, Inc., 2015. 2

[22] Cheng Tan, Zhangyang Gao, and Stan Z. Li. SimVP: Towards Simple yet Powerful Spatiotemporal Predictive Learning. *arxiv*, 2211.12509, 2023. 2

[23] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021. 1

[24] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. PVT

v2: Improved baselines with Pyramid Vision Transformer. *Computational Visual Media*, 8(3):415–424, 2022. 1

[25] 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*. Curran Associates, Inc., 2017. 2

[26] Yunbo Wang, Haixu Wu, Jianjin Zhang, Zhifeng Gao, Jianmin Wang, Philip S. Yu, and Mingsheng Long. PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2208–2225, 2023. 2

[27] Yuxin Wu and Kaiming He. Group Normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1

[28] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical Evaluation of Rectified Activations in Convolutional Network. *arxiv*, 1505.00853, 2015. 1