Deep Learning-based Parameter Transfer in Meteorological Data

Fatemeh Farokhmanesh,[a] Kevin Höhlein,[a] and Rüdiger Westermann[a]

[a] *Department of Informatics, Technical University of Munich, Germany*

Corresponding author: Fatemeh Farokhmanesh, fatemeh.farokhmanesh@tum.de

ABSTRACT: Numerical simulations in earth-system sciences consider a multitude of physical parameters in space and time, leading to severe I/O bandwidth requirements and challenges in subsequent data analysis tasks. Deep-learning based identification of redundant parameters and prediction of those from other parameters, i.e. Variable-to-Variable (V2V) transfer, has been proposed as an approach to lessening the bandwidth requirements and streamlining subsequent data analysis. In this paper, we examine the applicability of V2V to meteorological reanalysis data. We find that redundancies within pairs of parameter fields are limited, which hinders application of the original V2V algorithm. Therefore, we assess the predictive strength of reanalysis parameters by analyzing the learning behavior of V2V reconstruction networks in an ablation study. We demonstrate that efficient V2V transfer becomes possible when considering groups of parameter fields for transfer, and propose an algorithm to implement this. We investigate further whether the neural networks trained in the V2V process can yield insightful representations of recurring patterns in the data. The interpretability of these representations is assessed via layer-wise relevance propagation that highlights field areas and parameters of high importance for the reconstruction model. Applied to reanalysis data, this allows uncovering mutual relationships between landscape orography and different regional weather situations. We see our approach as an effective means to reduce bandwidth requirements in numerical weather simulations, which can be used on top of conventional data compression schemes. The proposed identification of multi-parameter features can spawn further research on the importance of regional weather situations for parameter prediction, also in other kinds of simulation data.

1. Introduction

The rapid increase in available computing power has enabled a broad adoption of simulation-based research methodologies in earth-system sciences. Numerical simulations of spatio-temporal dynamical systems consider a multitude of physical parameters and are carried out at high resolution in space and time. To account for the uncertainty in the representation of certain physical processes, in meteorology and climate modelling numerical ensemble simulations are carried out with varying magnitudes of initial condition uncertainty. Simulations are performed routinely by weather centers worldwide, and in research we see increasing use of unique super-ensembles consisting of hundreds and even thousands of members (Necker et al. 2020).

In classical workflow scenarios, simulations are run on large-scale computing facilities and data are streamed to and stored on external file systems for archiving and subsequent analysis. However, the volume of generated data has reached an order of magnitude where the speed of data transfer between computing device and file system – so-called I/O operations – imposes a major bottleneck. For instance, over the last decade the ability to compute increased about two orders of magnitude on supercomputers, while the ability to store and load data only increased about one order of magnitude.

The divergence between compute and I/O renders the classical simulation workflow increasingly problematic and requires to avoid streaming or even simulating data that can be recovered from the generated results. When using data compression, this can significantly reduce the time it requires to bring the data to the compression stage (e.g., when using distributed memory architectures) and perform the compression.

Within this line of research, deep-learning-based Variable-to-Variable (V2V) transfer has been proposed recently by Han et al. (2021b) for optimizing information transfer in situations where spatio-temporal multi-parameter simulations can be carried out in far less time than it requires to store the data on a file system. V2V considers each simulated parameter as a separate entity and proposes an algorithm to identify groups of similar parameters and one representative member from which the other parameters in this group can be inferred.

V2V represents all simulated parameter fields in a common feature space (the so-called latent-space) that is learnt by a convolutional neural network (CNN), and identifies subsets of similar parameters in this space. For each subset the most representative member is determined, and another suitably trained network then learns to reconstruct all other member in one subset from the representative one. The bandwidth requirements for storing the multi-parameter data is reduced to the bandwidth required for storing the representative parameter fields and the weight parameterization of the reconstruction networks.

Contribution

In this work, we assess the applicability of V2V transfer to earth-system related data, and identify shortcomings of the proposed methodology. In consideration of these findings, we propose an improved approach, in which a single CNN is trained on meteorological archives to learn general relationships between subsets of parameters, thereby focusing on subsets of parameter fields with vastly different characteristics and variation in 2D space and time. We demonstrate the capabilities of the proposed approach using two exemplary datasets, representing different modalities of meteorological data. We consider a global reanalysis dataset, which is taken from the WeatherBench (WB) benchmark suite (Rasp et al. 2020), and study the performance of the proposed approach on data on large spatial scales. Furthermore, we apply the proposed method to an ensemble forecast dataset, which was generated by Necker et al. (2020) to study the sampling accuracy of spatio-temporal correlation patterns in convective-scale forecast ensembles. Fig. 1, as well as Fig. B1, demonstrate significant structural variability between the single-parameter fields in each dataset. Fig. 1 (a) shows snapshots of the parameter fields at a particular point in time. Fig. 1 (b) shows a two-dimensional embedding (computed with t-SNE, van der Maaten and Hinton 2008) of the V2V latent-space representations of these fields at different time points, which are used to search for parameter similarities. Fig. B1 displays the same overview for the convective-scale ensemble (CSEns) dataset. It can be seen that visible clusters involve only a single parameter, and clusters of distinct parameters are situated at roughly the same distance from one another. The lack of similarities between pairs of parameters prohibits the use of the original V2V algorithm.

Based on these observations, we propose a different strategy for V2V transfer, which considers the expressiveness of subsets of multiple parameters instead of transferring only between pairs of them. To identify these subsets, a CNN-based model architecture is trained multiple times on multi-parameter fields with varying parameter subsets removed from the input data. In an ablation study, we then shed light on the prediction skills of the different models, depending on

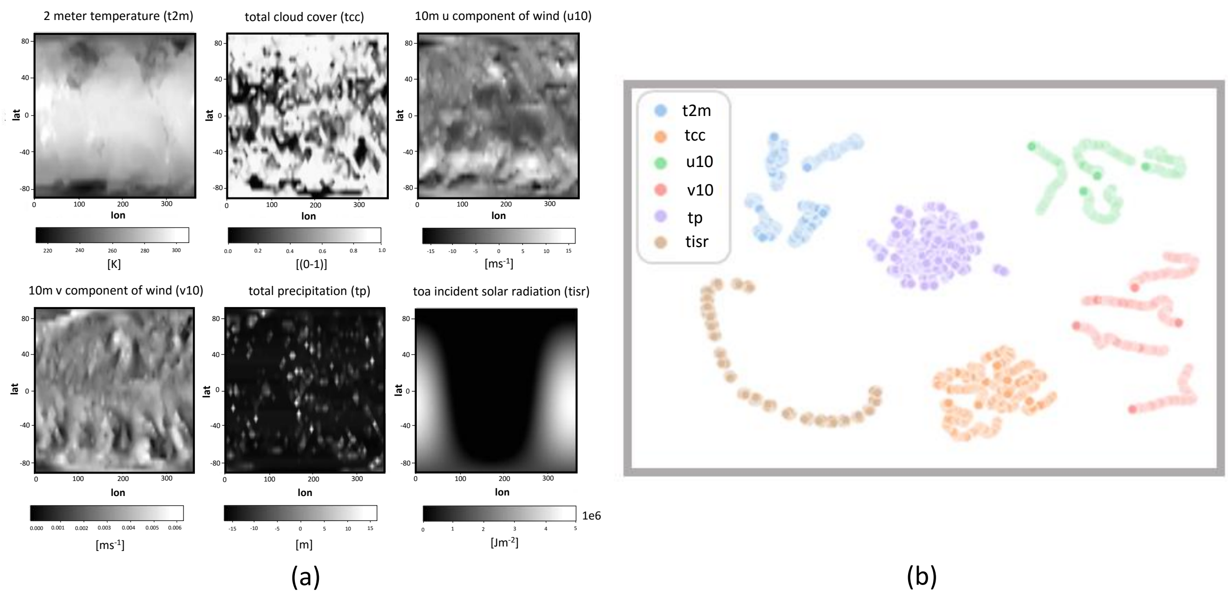


FIG. 1. Different parameter fields in the ERA5 reanalysis dataset. a) Gray-scale visualizations of the parameter fields at a particular time. b) t-SNE projections of latent-space features of the parameter fields (different parameters indicated by colors) at different times (note that projections for different initializations of t-SNE yield similar groupings).

the parameter subsets that are used as source. For $n \in \mathbb{N}$ originally available parameter fields, the networks are designed to learn mappings from m ($< n$) parameter fields (the input) to predict the remaining $n-m$ ones (the output). Doing so, the networks encode the inputs into a compact latent-space representation, in which relationships between the m input fields and the $n-m$ output fields are encoded. The networks are trained via standard backpropagation with a loss function, which measures the reconstruction accuracy for all $n-m$ parameters in common. We demonstrate, using numerical and visualization-based quality metrics, that the networks efficiently learn to reconstruct the unseen parameters, thereby not overfitting to the provided training samples but generalizing to simulation snapshots that haven't been seen before.

Due to the high computational complexity of training all $\binom{n}{m}$ different networks for reconstructing $n-m$ fields from m given fields, we propose a computationally less involved strategy and demonstrate its effectiveness for selecting the most representative members. Conceptually, this strategy builds upon removing iteratively those parameters that are most difficult to predict from the remaining parameters, simultaneously avoiding keeping redundant fields in the input. Furthermore, the networks' training behaviours are monitored and the convergence rates at early training stages after few epochs of learning are used as indicators of the difficulty of parameter transfer. For instance, for one of our use cases comprising nine different parameters and selecting four of them to predict the remaining five parameters, training requires only roughly six hours on a low-size deep learning cluster with six mid-size GPUs, equipped with 11 GB of graphics memory, each.

Beyond considering the networks purely as black-box models, we further try to gain insight into the multi-parameter relationships that are learned by the models. For this purpose, we employ an adapted version of layer-wise relevance propagation (LRP, Bach et al. 2015), a method for highlighting input areas and parameters, which are important for the reasoning process of the models. This offers the opportunity to uncover feature patterns in multi-parameter space, i.e. reoccurring parameter combinations, which are recognized as important for the network to achieve high accuracy, and analyse their correspondence to certain weather situations.

The remainder of this paper is structured as follows. In section 2, we review related work. In section **??**, we summarize the original V2V algorithm and highlight algorithmic shortcomings. Building up on these findings, we present our extended V2V approach in section 4. We introduce the example datasets used for subsequent experiments in section 3 and describe the network architectures used for the experiments in section b. The ablation study, as well as the LRP analysis of the models, are carried out in section 5. We conclude the paper in section 6.

2. Related work

In recent years, machine learning with powerful deep-learning architectures has found applications in various fields of climate science and meteorology (Reichstein et al. 2019). Many of the possible applications exploit the efficiency and flexibility of CNN architectures when applied to inference tasks involving grid-structured data.

a. Super-resolution and downscaling techniques

Related to our approach are so-called super-resolution and downscaling techniques, which reconstruct high-resolution parameter fields from corresponding low-resolution versions. In contrast to V2V approaches, the information transfer occurs between representations of the same parameter with different spatial resolutions. Some of these approaches can be used for data compression, in principle. Such methods operate by first sub-sampling the parameter fields and subsequently reconstructing the initial fields from the sub-sampled versions. For example, Rodrigues et al. (2018) proposed a supervised convolutional neural network that interpolates a low-resolution weather data into a high-resolution output. Pouliot et al. (2018) introduced deep learning-based enhancement in landsat super-resolution. Cheng et al. (2020) proposed a method that converts low-resolution climate data to high-resolution climate forecasts using Laplacian pyramid super-resolution networks. Downscaling approaches, in contrast, aim at predicting additional high-resolution details from low-resolution parameter fields, without assuming prior knowledge of the original high-resolution data. For instance, Höhlein et al. (2020) and Serifi et al. (2021) train on a small set of paired low- and high-resolution simulation pairs to circumvent generating expensive high-resolution simulations at inference time at all. These techniques, even though they establish relationships between low- and high-resolution fields, are not motivated by the idea to compress the data.

Super-resolution of scientific data has also been investigated from the perspective of scientific visualization, since high-resolution simulations in meteorology make analysis of these datasets challenging. Early works on data super-resolution demonstrate the capabilities of neural networks to learn upscaling a low-resolution version of the data to the initial high-resolution dataset. Upscaling is performed in the spatial domain (Zhou et al. 2017; Han and Wang 2020; Guo et al. 2020), in the temporal domain (Han and Wang 2019), and in the spatio-temporal domain (Han et al. 2021a). Underlying these works is the goal to avoid storing the high-resolution datasets and, thus, reduce bandwidth and memory requirements. By training networks to infer the full image from a low-resolution image of an iso-surface, Weiss et al. (2019) demonstrate improved rendering frame rates. In recent work by Weiss et al. (2020) a convolutional neural network learns to adaptively place image samples and reconstruct the full image from the generated unstructured set of samples.

To work in situations with severe I/O limitations, Sato et al. (2019) introduced a so-called in-situ approach for visualizing and post-processing high-resolution meteorological data. In-situ approaches process the data instantly when it is produced by the simulation, without involving storage resources. Röber and Engels (2019) analysed in-situ data processing approaches in climate science. Helbig et al. (2015) proposed a visualization workflow where the first stage is a data abstraction layer that downsamples the data spatially and temporally. Toderici et al. (2017) proposed an image compression method using a recurrent neural network by saving the compressed latent space produced by the network instead of the high-resolution data.

A different approach for data compression has been introduced by Han et al. (2021b) for multi-parameter data, by training networks to infer certain parameters from others, and, thus, to avoid storing these parameters. Conceptually, this work builds upon the notion of information transfer between scalar fields (Wang et al. 2011) to derive transferable parameter pairs.

b. Variable-to-variable (V2V) transfer

The overall goal of V2V lies in identifying those variables in a multi-parameter dataset that can be redundantly reconstructed given other parameters from the same dataset. Han et al. (2021b) subdivide the V2V process into 3 conceptually distinct stages: feature learning, translation graph construction and variable translation. First, a CNN model architecture such as UNet (Ronneberger et al. 2015) is trained in an auto-encoder-like setting to encode and reconstruct snapshots of parameter fields. The same model is shared between all parameters, such that the internal hidden variables of the model, referred to as the latent-space representation, are informed about similarities and dissimilarities between different parameters. After training, all parameter snapshots are mapped into the latent space where clusters of similar parameters are detected. Han et al. (2021b) propose to find clusters through visual examination of the latent-space features. To visualize the features, they apply a non-linear dimension reduction algorithm, called t-distributed stochastic neighbor embedding (t-SNE, van der Maaten and Hinton 2008). Parameters in the same cluster become a transferable variable group.

In the translation-graph construction stage, the Kullback-Leibler divergence is used to estimate a measure of so-called *transferable difficulty* for pairs of parameters inside a transferable parameter group. Parameter pairs are considered for transfer, if the Euclidean distance between their respective latent-feature representations is smaller than a predefined distance threshold. Parameters in different transferable parameter groups are not considered for transfer. A directed transfer graph is then constructed by chaining transferable pairs according to a minimum discrepancy criterion. Finally, in the variable transfer stage, CNNs are trained to learn the transfer mapping according to the translation graph.

V2V reduces the search for transferable variables to pairs of similar variables based on a distance threshold criterion and the visual analysis of a t-SNE projection. This leads to a number of shortcomings regarding the expressiveness of the identified parameter relations and reproducibility of the results. V2V does not assess true predictability of one variable based on another, but uses an empirically determined approximate criterion, which might overlook potentially valuable relationships when the threshold value is not set optimally.

Furthermore, V2V cannot always decide unambiguously the source and target fields. This decision is based on the translation graph where the nodes correspond to the parameter fields and directed edges indicate transferability from a source to a target variable. In this graph, however, cycles can occur. This happens because the *transferable difficulty* is based on the Kullback-Leibler divergence, which is not a metric and does not satisfy the triangle inequality in general. When only pairwise transferabilities are considered, this can result in the selection of all parameters in a set of similar parameters as sources and targets of one another, respectively.

3. Datasets

We validate our approach with the WeatherBench dataset, which has been proposed as a benchmark dataset for data-driven, medium-range climate prediction problems (Rasp et al. 2020), and the convective-scale forecast ensemble generated by Necker et al. (2020) (see appendix B).

In both cases, the proposed neural network models receive as input an array of shape $m \times H \times W$, with m the number of physical input parameters, and $H \in \mathbb{N}$ and $W \in \mathbb{N}$ denoting the spatial dimensions of the field. Assuming an initial number of $n \in \mathbb{N}$ physical parameters, the model output is a field of shape $(n-m) \times H \times W$, which contains reconstructions of the parameters that have not been considered in the input (see Fig. 2).

WeatherBench (WB) is based on ERA5 atmospheric reanalysis data (Hersbach et al. 2020) generated regularly at the European Center for Medium-Range Weather Forecasting (ECMWF) through data assimilation procedures, combining spatio-temporal numerical simulations and observation data. To facilitate the accessibility to machine learning workflows and accelerate studies in weather prediction, WB provides regridded ERA5 reanalysis data on regular latitude-longitude grids with three different resolutions and 13 different pressure levels. The data is available hourly for 40 years from 1979 to 2018.

We consider a selection of 2D single-level fields with a resolution of 1.40525° in latitude and longitude, resulting in a domain size of 128×256 vertices for global data. The selected physical parameters are 2 m-temperature (t2m), total cloud cover (tcc), u- and v-component of 10 m-wind (u10, v10), total precipitation (tp), and top-of-atmosphere incident solar radiation (tisr). As in previous studies (e.g., Höhlein et al. 2020), where the prediction accuracy of convolutional neural networks could be improved by using orography information, orography height is added as an additional constant predictor. To facilitate model training, all field values are standardized before training the models. This normalization scheme enforces equal variation in all parameter fields under consideration, which is helpful for ensuring comparability of reconstruction accuracy metrics. We utilize two different global and local standardization. In local standardization, rescaling is computed for each grid location from the statistics of all time steps, while in global standardization, mean and standard deviation values are computed over the whole domain for all time steps. Global standardization performs better in our case and helps to reduce the reconstruction error. We refer this to the fact that the local standardization can enhance uniformity of the data, but destroys spatial coherence patterns. We use the 23 first years of WB during training. Of these, 20 years serve as training data for fitting the models, and three years are reserved for validation. The remaining years are left out for testing and visualization.

4. Method

To overcome the shortcomings of V2V, we propose an alternative parameter selection procedure. It replaces the single-parameter auto-encoding CNN and subsequent clustering and pairwise similarity search with multi-parameter CNNs and loss-based tracking of the learning progress. Initially, given a multi-variate time-varying dataset with $n \in \mathbb{N}$ parameters and $T \in \mathbb{N}$ timesteps, the user selects the number m of input fields from which the remaining $n - m$ parameter fields are predicted.

a. Parameter selection

The most straight forward – yet computationally demanding – approach is to launch $\binom{n}{m}$ training runs for the different parameter configurations, and select the network with the lowest loss. By starting from $n - 1$ inputs and one output and proceeding iteratively with decreasing number of inputs, the procedure can also be made dependent on a predefined loss threshold, i.e., by launching for all k, $1 \leq k \leq m$, a batch of $\binom{n}{k}$ training runs and stopping once the minimum loss exceeds a given threshold. The parameter configuration for which the minimum loss is achieved is then selected for variable transfer. In our current implementation we consider only the user-specified number of input parameters.

Since for large values of n, the described procedure requires training too many networks, a computationally less expensive alternative needs to be developed. A straight forward approach is to train only n networks, where each network predicts one single parameter from the remaining $n-1$ parameters, and use the networks' losses as indicators of how difficult the prediction of a parameter is. If a loss value is high, predictability is low, and, thus, the parameter predicted with the highest loss is fixed as one of the input parameters. Then, of all trained networks where this parameter is already contained in the input set, the one with the largest loss is selected, and the variable that is predicted is added to the input. This procedure is repeated until the specified number of inputs is reached.

In our experiments, this approach finds exactly the same input and output sets as the exhaustive training procedure, yet it requires training only n networks. In general, however, since only the difficulties of predicting each parameter individually from all other parameters are considered, parameter combinations with higher prediction strength can be overlooked. For instance, consider a subset of similar parameters, each of which can be predicted at high accuracy from the other parameters in this subset. In this situation, the loss values of the reconstruction networks will be low, and it becomes unlikely that one of these parameters will be selected as input. Consequently, either the input solely comprises parameters from which the ones in the subset cannot be well predicted, or m needs to be so large that of each subset of similar parameters at least one is selected. However, since parameters from the most similar subset will be considered last, m can become too large to be of any practical relevance.

To address these shortcomings, we propose a strategy with lower computational complexity than the first strategy, and which differs from the second strategy in that it considers subsets of parameters in both the inputs and predicted outputs. As in the previous strategy, n networks are trained initially, with each network predicting a single parameter from the remaining $n-1$ parameters, and the parameter predicted by the network with the highest loss is fixed in the input.

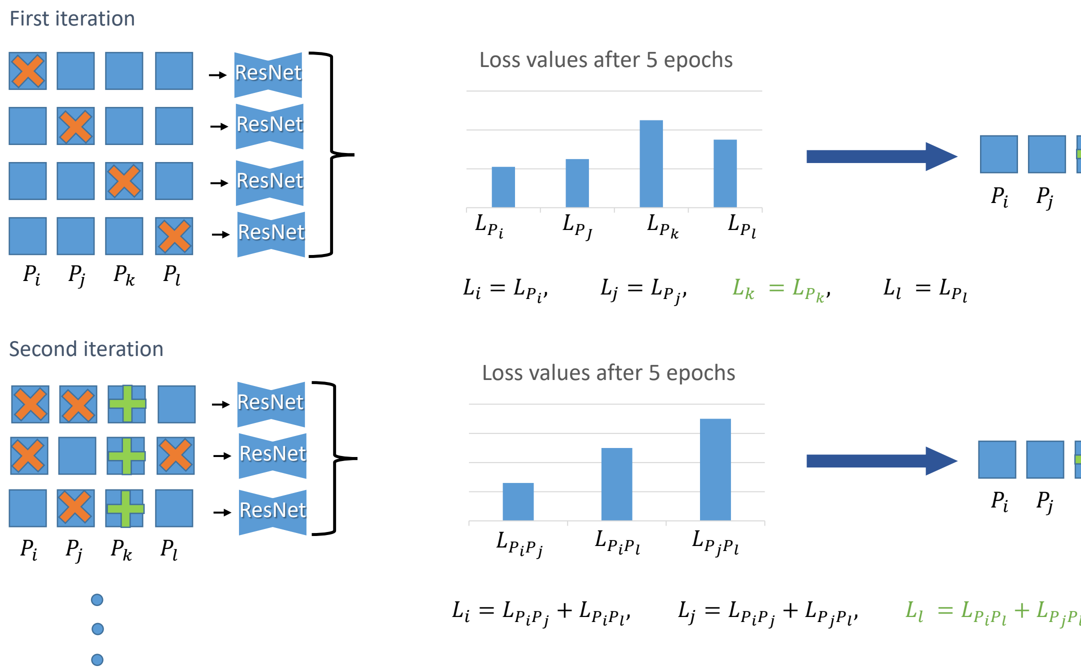


FIG. 2. Method overview. In the i-th iteration, the same Resnet is trained multiple times using different combinations of $n-i$ input and i output parameters. Orange crosses indicate the output parameters, green pluses indicate the parameters that are fixed in the input set, blue squares indicate the remaining parameters in the input set. In each iteration, for each free parameter a loss is computed by adding the losses of those networks in which the parameter is in the output set. The parameter with the maximum loss is fixed in the input set. For the WB dataset, orography is used in every input, but is not predicted.

In the next iteration, all networks with $n-2$ inputs (including the fixed parameter) and two outputs are trained. For all but the fixed parameter, the overall loss is computed by adding the losses of all networks where this parameter is in the predicted set. The parameter with the highest loss is fixed, and the procedure moves on with $n-3$ inputs, and three outputs now containing the two fixed parameters (see Fig. 2 for a graphical overview of the proposed approach). This strategy, in case of a similar subset of parameters, recognizes when a certain output cannot be well predicted and then fixes an input which is necessary to achieve higher accuracy.

In our experiments, all parameter fields are normalized before training to equilibrate differences in parameter magnitudes and variation. If the dataset contains static fields like orography, these fields are concatenated to the inputs to serve as additional information for the models. In particular, orography is used with the WB dataset to enable the networks learning dependencies between the parameters and land-/sea-scape, and, thus, enhance their inferencing skills. The quality of the reconstruction of all parameters is measured by a suitable loss function, e.g. L_1 loss, and the model weights are optimized using standard backpropagation. We monitor both the training and validation loss to avoid overfitting.

A network's loss curve indicates how difficult it is for the network to achieve an accurate reconstruction depending on the current input and output parameters. I.e., depending on which parameters are used, the reconstruction error decreases more or less quickly. While saturation of both losses typically happens after 70 epochs, our experiments show that already after few epochs of training the reconstruction error clearly reveals the differences between different parameter combinations. In particular, when comparing the loss curves in these early stages with the loss curves after convergence, the relative behaviour of the networks does not change. This indicates that network training does not need to be performed until convergence, but can be stopped after few epochs to obtain an indication of the reconstruction quality. In particular, we consider loss values after five epochs of training, resulting in roughly one hour (for training 20 networks) on a low-size deep learning cluster with six mid-size GPUs to determine three input and three output parameters for the WB dataset, comprising six parameters. For the CSEns dataset, comprising nine different parameters, the proposed procedure requires roughly six hours for training 87 networks to determine the four input parameters that best predict the remaining five output parameters.

Compared to the V2V approach by Han et al. (2021b), the proposed strategy is computationally more expensive, yet it exhibits a number of advantages: Firstly, we obtain a more accurate measure of transferability, since our models are directly trained to reconstruct parameters. Second, the proposed approach is not constrained to selecting pairs of parameters, but can uncover multi-parameter relationships. Lastly, the method, in principle, enables to set a loss threshold for triggering the stopping of iterations. Due to normalization of the target parameters, this threshold can be interpreted as a measure of acceptable relative error, and is thus more accessible than the distance threshold in the latent-space features, which was considered in the original V2V algorithm.

b. Network architectures

In this study, we propose to train a deep convolutional neural network (CNN) architecture to predict a certain number of output parameters from a given set of input parameters. In general, deeper networks can have higher prediction quality, yet they can easily lead to convergence problems in the optimization process due to vanishing gradients (Glorot and Bengio 2010). In early layers of the network, gradient estimation causes an exponential decay of the gradient magnitudes, so that the parameters cannot change significantly in the training process. An efficient way to overcome this problem is to utilize short-cut or residual connections as in ResNet architectures (He et al. 2016). In such architectures, outputs of earlier layers are added to the output of later layers, thus circumventing the accumulation of intermediate gradients.

In this study, we select a ResNet architecture with three residual blocks. A schematic representation is shown in Fig. 3. The input block of the model consists of a single convolution layer with kernel size of $(3,3)$, 64 channels, batch normalization, and leaky rectified linear unit (LeakyReLU). We use input padding before each convolution layer. To account for periodic boundary conditions in the longitude direction of the WB dataset, we employ a periodic padding scheme in this dimension, and replication padding elsewhere. After that, there are three residual blocks and each block has two convolution layers with 64 channels and kernel size $(3,3)$. Since the number of parameters grows with the kernel size, it is cost efficient to select a kernel of size 3. After the first convolution layer in the residual block, the network utilizes a batch normalization layer, a LeakyReLU layer, the second convolution layer, and another batch normalization layer. Batch normalization is used to achieve improved stability and convergence (Ioffe and Szegedy 2015). After each residual block, a LeakyReLU activation function guarantees non-linearity of the mapping. The final layer is a single convolution layer with kernel size of $(3,3)$ and $n-m$ output channels.

As an alternative to the ResNet architecture, we also analysed the potential of a UNet architecture (Ronneberger et al. 2015) for loss-based parameter selection.

In contrast to the ResNet architecture, which operates on a single spatial scale throughout the whole architecture, the UNet architecture allows for the extraction of features on multiple spatial scales, which offers the possibility to learning a wider range parameter relationships. The UNet consists of two symmetric branches, which give it the characteristic u-shape, as seen in Fig. 4. In the encoding branch, the data are encoded into an abstract reduced feature representation, and in the

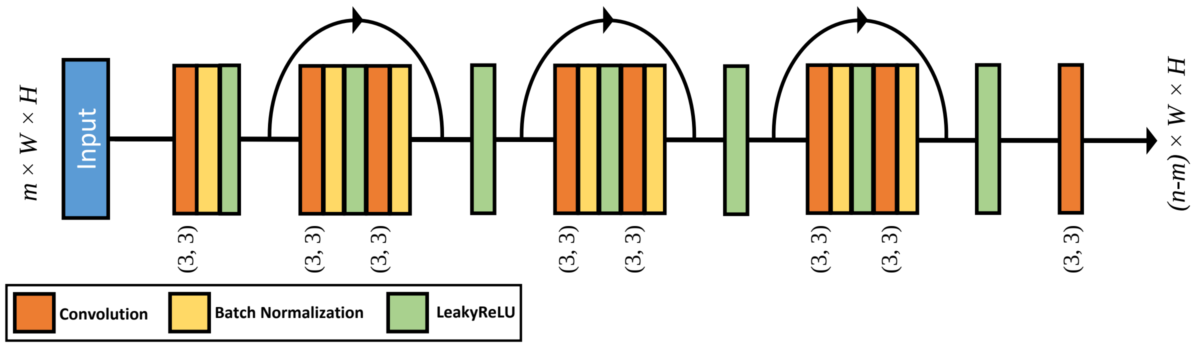


FIG. 3. Schematic of the used ResNet architecture. It consists of three residual blocks, each followed by a LeakyReLU activation function. m ($< n$) input fields are fed into the input convolution layer with 64 channels and kernel size of (3, 3), a batch normalization layer, and LeakyReLU activation function. The network comprises three residual blocks and a final single convolution layer. The network predicts $n - m$ output fields.

decoding path the feature representations are then decoded to reconstruct the predicted fields at the target resolution. During the encoding step, the resolution is iteratively reduced, and the number of feature channels is increased at the same time. In the decoding step, while reducing the number of feature channels, the features are super-sampled to a higher resolution. The paths are connected by skip connections, which concatenate feature channels from the encoder with corresponding features from the decoder, in order to precisely preserve and localize the information in the data that could be lost in the encoding stage. The most bottom layer of the UNet, i.e., the bottleneck layer, enforces the model to learn a compact representation of the input containing the globally most relevant information to recover it.

In our experiments, the UNet architecture did not improve the reconstruction quality significantly, yet increased the training time due to its higher computational complexity. A sample of the reconstruction quality of the UNet architecture is shown in Fig. A2. Nevertheless, we found that ResNet and UNet seem to learn different mappings internally, which we discuss in more detail in section 5 c.

The presented architectures have been designed through empirical experimentation, trading of model flexibility and reconstruction quality against applicability to diverse datasets and computational efficiency. Especially for data fields on spherical geometries, more sophisticated network designs exist, see, e.g., the survey by Cao et al. (2020). Such architectures, however, come at higher

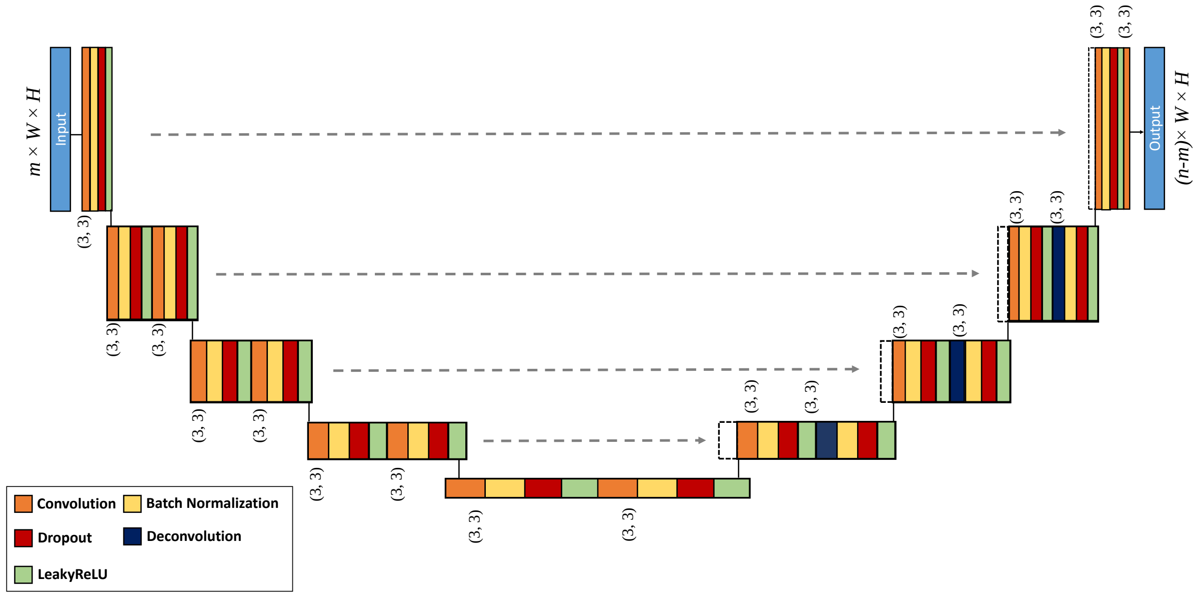


Fig. 4. Schematic of the used UNet architecture. The contracting branch is comprised of three convolutional blocks, each consisting of two convolution layers with subsequent batch normalization, dropout, and a LeakyReLU activation function. The expansive branch includes three deconvolution blocks, each consisting of one convolution, and one deconvolution layer. Each of these layers is followed by a batch normalization layer, dropout, and LeakyReLU activation function. m ($< n$) input fields are fed into the input block, which contains a single convolution layer with 64 channels and kernel size of (3, 3), followed by batch normalization layer, dropout, and LeakyReLU activation function. The number of reconstructed fields is $n - m$.

computational complexity or require careful data-specific selection of hyper-parameters to achieve better performance than standard CNNs, and are thus not considered in the present study.

The error between target fields and predictions is measured in terms of L_1 distance, which we prefer over L_2 due to empirically less pronounced suppression of outlying predictions.

5. Experiments

In an exhaustive ablation, we demonstrate the feasibility and reliability of our approach using the WB reanalysis dataset as a use case. The results of applying the proposed strategy to the CSEns dataset are shown in the appendix.

Via this ablation study, we aim to answer the following questions:

1. Which is the minimal set of input parameters from which the remaining parameters can be reconstructed accurately? This number indicates how aggressively the initial parameter set can be reduced.

2. Over which geographic regions do parameters strongly affect the network's prediction quality?

To answer the first question, we use loss-based parameter selection via the ResNet architecture. Both architectures are used subsequently to answer the second question by visualizing the sensitivity of the local prediction accuracy to regional changes of the input parameters.

a. Validation of the extended V2V approach

To validate the reliability and reproducibility of the proposed loss-based parameter selection, we train networks for different parameter configurations multiple times with different random weight initializations, and compare the order of the observed losses after five training epochs. For brevity, we first show results only for the case $m = 1$, which results in six different parameter configurations. For every configuration, we train 10 models and sample model ensembles by randomly picking one of the 10 models for each configuration. For each sample, we then rank the model configurations according to the observed loss value after five training epochs, and assess the consistency of the ranking order among different samples. Fig. 5 illustrates the observed loss statistics. We find that the separation in loss magnitude between different parameter configurations is typically larger than the variance of losses for each configuration (see Fig. 5, left). As a result, the ranking of losses is consistent between different runs. This is seen in the heat chart in Fig. 5 (right), which visualizes the frequency of how often a particular loss rank is observed for each of the parameters, and suggests an almost perfect one-to-one mapping between parameters and ranks. Both charts together confirm that our loss-tracking approach constitutes a reproducible criterion for selecting parameter configurations. Nevertheless, we observe that clustering of losses may occur, i.e. different configurations may result in very similar loss statistics (e.g., parameters tp, u10 and v10 in Fig. 5, left). Due to the overall small variation in losses per configuration, we conjecture that all of the possible outcomes are equally well-suited for further evaluations.

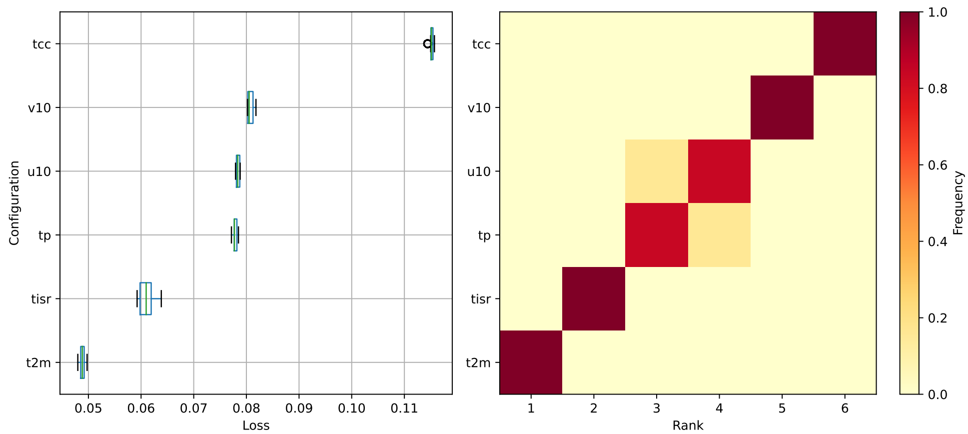


FIG. 5. Distribution and ranking of losses for different network configurations. Networks are trained for different parameter configuration ($m = 1$, i.e. one parameter is left out and is to be predicted) with different weight initializations. Left: Box plot of the loss statistics. Right: Heat map of the observed ranking order.

b. Ablation study

For the WB data comprising six parameters, we start with applying the loss-based selection procedure to predict one masked out parameter using the remaining five input parameters. This number is then increased to two and finally three predicted parameters, with four and three input parameters, respectively. This means that during the first iteration six networks, then $\binom{5}{2}$ networks and finally $\binom{4}{3}$ networks are trained. We do not go beyond three masked out parameters, since significantly reduced reconstruction quality is observed in this case.

To justify our decision to use the network losses after five epochs as indicators of the difficulty to predict a certain parameter or parameter combination, we analyze the loss values for five and 70 epochs of training of all networks that were trained. Fig. 6 shows the losses of all networks trained for five epochs by the proposed loss-based selection approach (top and bottom left charts), and the losses of all $\binom{6}{3}$ possible networks trained for five epochs (dark blue bars in bottom right). The loss values indicate that the loss-based selection approach finds the parameter combination yielding the lowest loss. Note that this is also confirmed for the CSEns dataset, as shown in Figs. B2 and B3 in the appendix. As shown by the overlayed loss values of the networks trained for 70 epochs (green bars in bottom right), training for five epochs shows very similar relative differences between different parameter combinations. Also this result is confirmed by the comparison of the

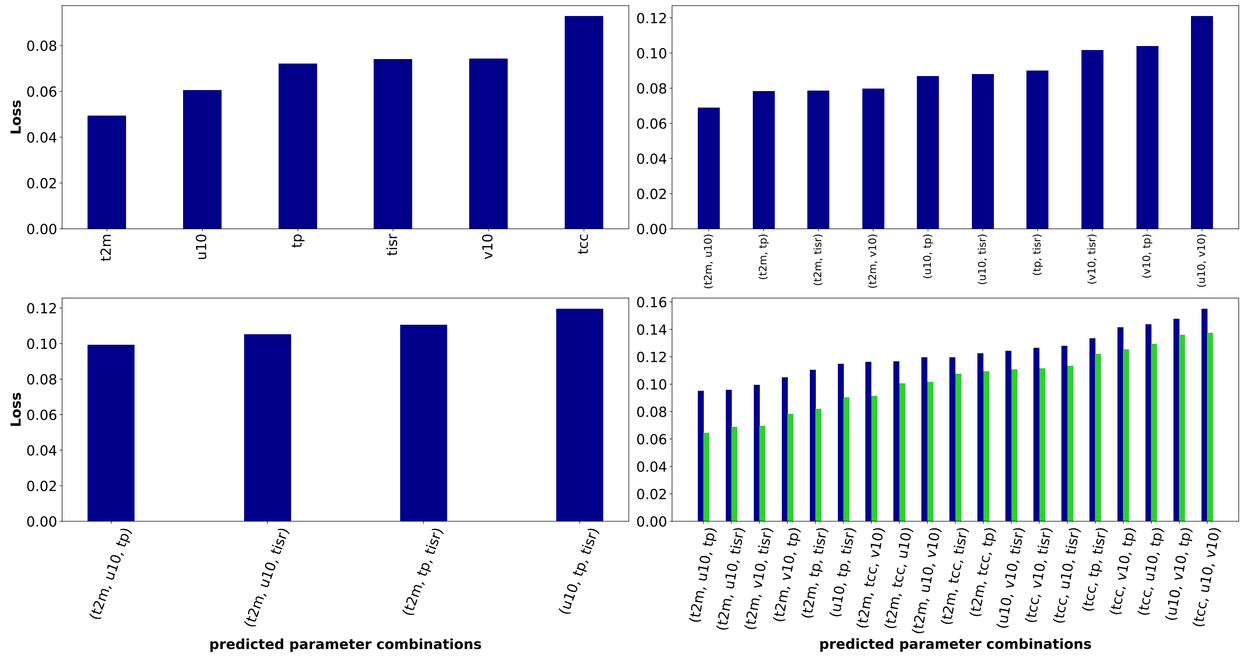


Fig. 6. Bar charts showing the losses of all networks trained for 3-to-3 parameter transfer with the WB dataset using the proposed iterative loss-based approach (top left: first iteration, top right: second iteration, bottom left: third iteration), and of all possible $\binom{6}{3}$ networks (bottom right). Blue bars represent losses after five epochs of training, green bars indicate losses after 70 epochs.

loss values for five and 70 of the CSEens dataset. When only one parameter is predicted from the remaining five parameters (plus orography), it can be seen that 2 m-temperature and total cloud cover, respectively, are the parameters that are easiest and most difficult to reconstruct. Thus, total cloud cover is the first parameter that is fixed in the input.

Fig. 7 shows the initial parameter fields (including orography) and the reconstruction results of the three parameters that have been masked out by the loss-based procedure. For comparison, the reconstruction results of the three worst parameter combinations are shown in A1 in the appendix. Notably, when all $\binom{6}{3}$ parameter combinations are evaluated, the very same combination is determined.

In Fig 8, for the selected parameter combination the resulting pixel-wise differences between the reconstructions and the initial parameter fields are shown. The quality of the reconstructed fields is measured using the image statistics SSIM (Wang et al. 2004) and the peak signal to noise ratio (PSNR), with the initial parameter fields as references. It can be seen that even when one half of

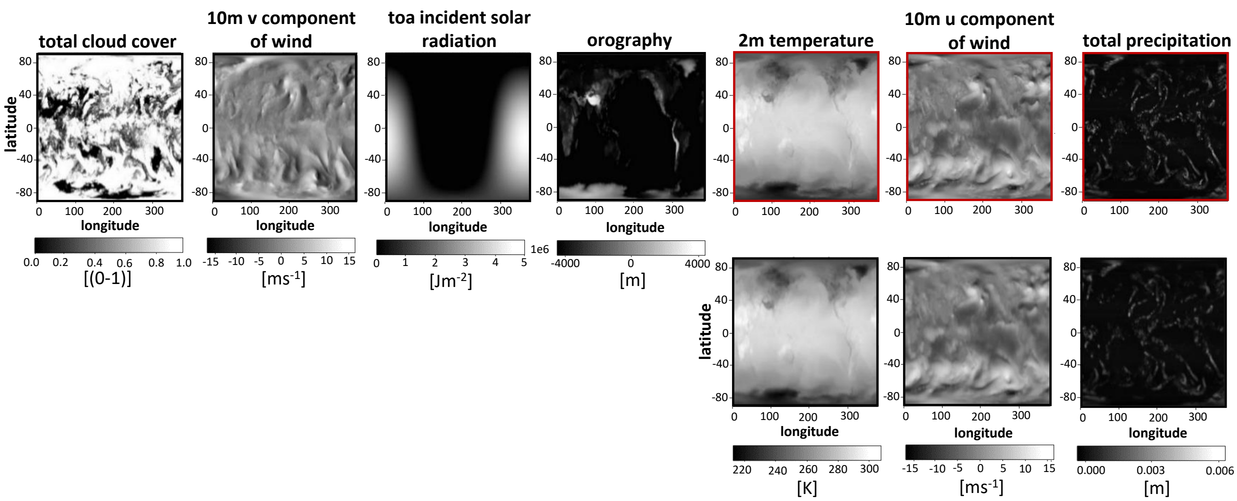


FIG. 7. Reconstruction results for the WB dataset when the network is trained to predict three parameter fields from three input fields and orography. Top: The initial parameter fields. A red outline indicates those fields the network has learned to predict from the others. Bottom: Predicted parameter fields.

the parameters are masked out, they can still be reconstructed at high accuracy by the network. In addition, the reconstruction quality that is achieved by the worst parameter combination is shown, i.e., the parameter combination yielding the highest loss of all possible $\binom{6}{3}$ parameter combinations. The results indicate the importance of a suitable procedure for finding the best parameter combination. The pixel-wise error plots indicate significantly different reconstruction quality between the best and worst parameter set.

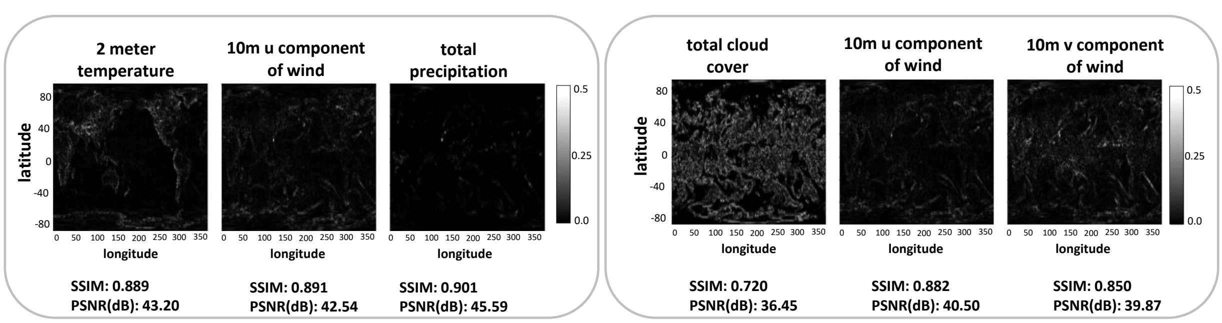


FIG. 8. Pixel-wise differences between the the initial and predicted fields when using the best (left) and worst (right) parameter combination. Per-pixel values are scaled by a factor of 10 for better visibility. Corresponding SSIM and PSNR (dB) values are given below each image.

c. *Feature analysis*

While the potential of the selected network architecture for V2V can be concluded from the results of the ablation study, no information can be drawn about what kind of dependencies are exploited by the networks. To shed light on this aspect, we use layer-wise relevance propagation (LRP) to localize the sensitivity of the reconstruction results to changes in the input parameter fields.

In its original form, LRP has been introduced as an explainability algorithm for image classification models (Bach et al. 2015), which is achieved by combining neuron activations and back-propagated gradient information to highlight image regions that exert a strong effect on the classifier output. LRP, thereby, builds on the concept of pixel-wise decomposition of the classifier score. I.e., given a classification score mapping of the form $f : \Omega \to \mathbb{R}$, where $\Omega \subseteq \mathbb{R}^{m\times H\times W}$ is the input domain (e.g., the space of images with $H \times W$ pixels and m channels per pixel), and $f(x) > 0$ (< 0) indicates evidence for presence (absence) of a particular feature, LRP attempts to find a set of relevance values $R_{kij} \in \mathbb{R}$ associated with the pixel values, such that the classification score can be approximated as

$$f(x) \approx \sum_{k=0}^{m-1} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} R_{kij}, \tag{1}$$

and $R_{kij} > 0$ (< 0) indicates that pixel channel k at position (i, j) contributes evidence in favor of (against) the presence of the feature in question. In the case of deep neural networks, which are composed of linear transformations with element-wise activation functions, suitable relevance values can be computed via iterative relevance back-propagation, subject to propagation rules (Bach et al. 2015).

Deviating from the setting of standard LRP, the input of V2V models is not an image, but a multi-dimensional array, representing a multi-parameter field, and the output is not a univariate classification score, but a multi-dimensional multi-parameter field. The difference in input modalities is only of limited importance, since the multi-parameter fields can be interpreted directly as multi-channel images. However, the complexity of the model output prevents straight forward application of standard LRP. We therefore propose to use an adapted variant of LRP to gain insight into spatio-temporal relevance and correlation patterns between model predictions and inputs.

Given a model mapping of the form $f : \Omega \rightarrow \mathbb{R}^{(n-m)\times H\times W}$, we propose adding an additional selector layer $s : \mathbb{R}^{(n-m)\times H\times W} \rightarrow \mathbb{R}$ at the end of the model, such that the output of the combined model, $s(f(x)) \in \mathbb{R}$, admits an additive decomposition according to Eq. (1), and can thus be further analyzed using standard LRP. Possible choices for s include summation operators, such as global (or local) averaging of field values or deviation measures, or selection operations, which select single pixels and output channels for computing LRP relevances. Depending on the choice of selector layer, different aspects of the input-output relationship can be investigated. For instance, a selector function returning the mean value of the output channel $0 \leq c < m$ inside a region defined by the pixel set $I \subseteq \{(i, j) : 0 \leq i < H, 0 \leq j < W\}$, i.e.

$$s_I^{(c)}(x) := \left\langle [x]_{kij} \right\rangle_{(i,j)\in I, k=c}, \tag{2}$$

where $[x]_{kij}$ denotes selection of the element at position (k, i, j) in the array x, yields positive relevance for input regions. This causes an increase of the averaged quantity according to Eq. (1). In contrast, functions of the form

$$\delta_I^{(c)}(x; x_0, p) := \left\langle \left|[x - x_0]_{kij}\right|^p \right\rangle_{(i,j)\in I, k=c}, \tag{3}$$

$p > 0$, yield positive relevance for regions which increase the deviation between the model prediction x and a certain reference prediction x_0 within the region I.

Figs. 9 and 10 show relevance maps for the global atmospheric situation on May 15, 2004, 08h, as seen in the WB dataset, using the ResNet (Fig. 9) and the UNet model (Fig. 10). We employ an absolute-difference-based selector function with a focus on single pixel deviations of the predicted quantities from the respective target value, i.e. $\delta_I^{(c)}(x; x_0, 1)$ with $I = \{(i^*, j^*)\}$, $0 \leq i^* < H$, $0 \leq j^* < W$, and x_0 denoting the target field. This allows drawing information about what parts in the data push the separate prediction channels away from the actual target value. Figures are shown for $(i^*, j^*) = (63, 127)$, which corresponds to the center pixel in the image, located at $0°N$ $180°E$. Relevance maps for other dates and pixel indices look similar. All output channels are treated separately, yielding a matrix of relevance maps, which visualize relationships between channel-wise prediction errors and model inputs. The back-propagation of relevance

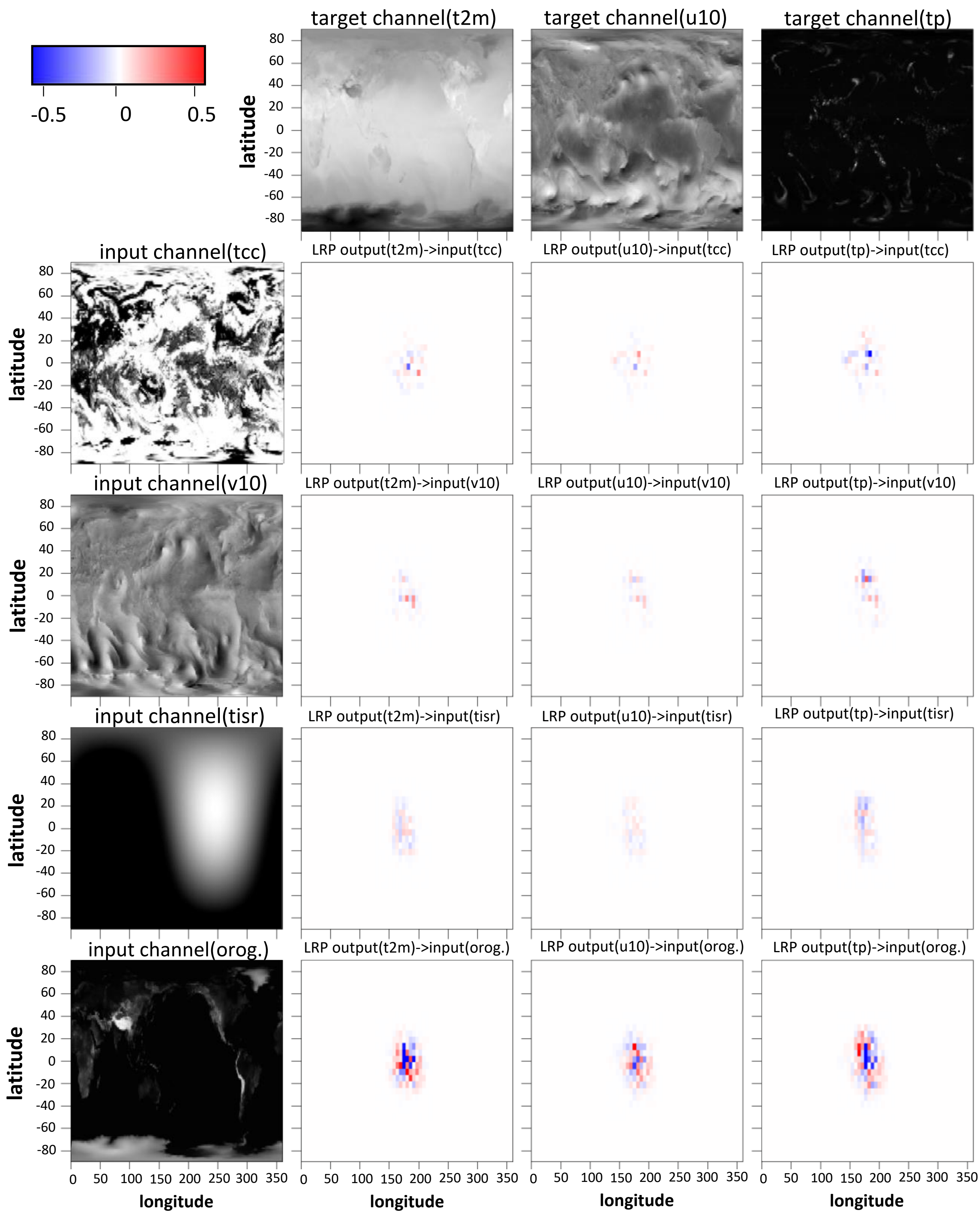
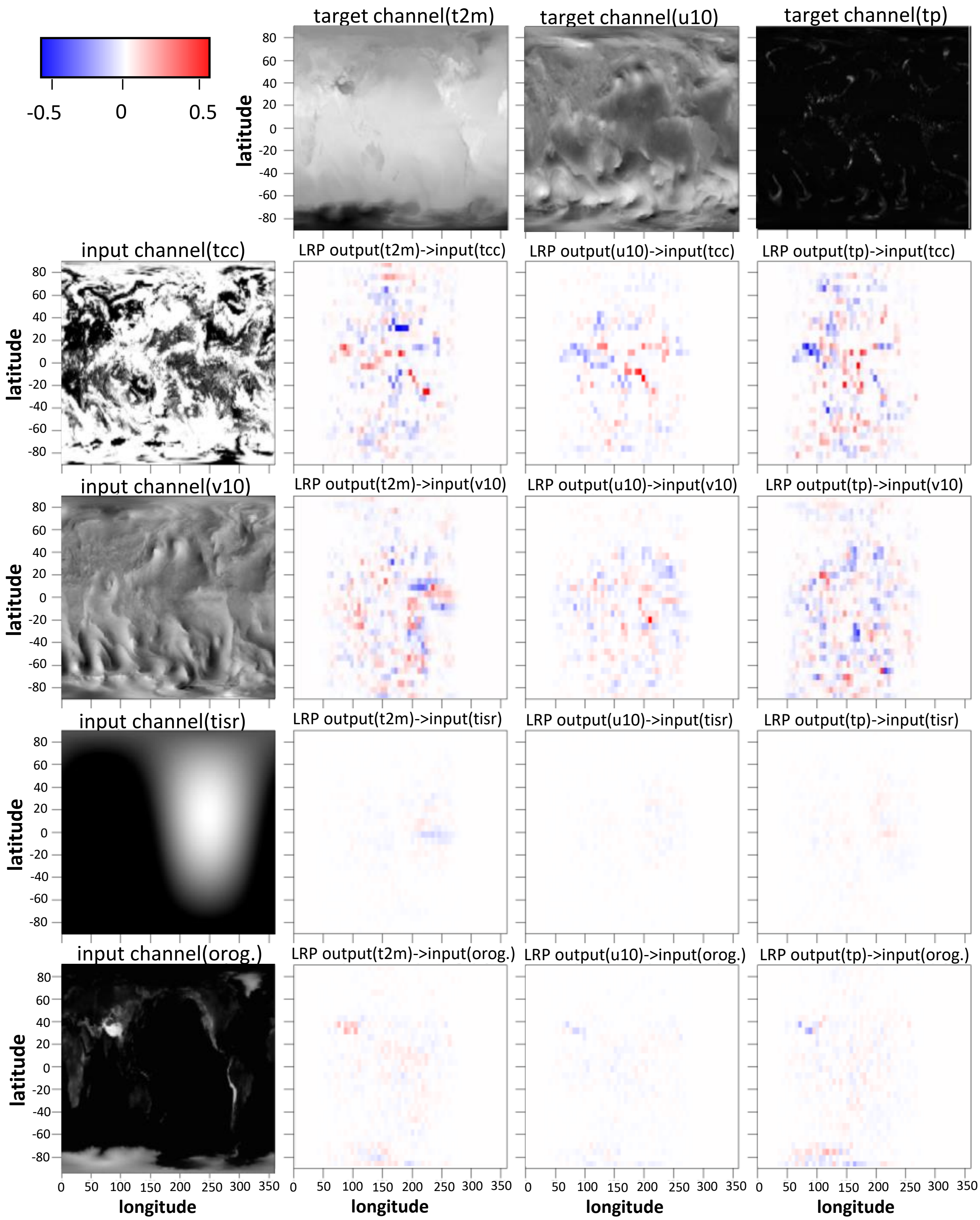


FIG. 9. LRP relevance maps with deviation-based selector function for the ResNet model in the best input-output configuration wrt. the proposed selection procedure. Timestamp of data sample: May 15, 2004, 08h.



503 FIG. 10. LRP relevance maps with deviation-based selector function for the UNet model in the best input-output configuration wrt. the proposed selection procedure. Timestamp of data sample: May 15, 2004, 08h

values according to our proposed feature selection is carried out using the standard LRP algorithm, which is available in the Captum model interpretability library for Pytorch (Kokhlikyan et al. 2020).

While the relevance patterns appear noisy in both cases, the structure of the relevance maps differs significantly between the model architectures, despite being trained on the same task and with the same set of training data. The ResNet architecture favors relevance distributions which are concentrated around the reference location. This is consistent with the inductive bias of the architecture, which arises from the use of convolution layers with small kernel sizes (see section b). Positive and negative relevances appear to be distributed randomly throughout the map, but significant differences are observed in the magnitude of the relevances. Relevance values wrt. orography possess larger magnitude in both positive and negative orientation than the remaining parameters. For the cloud-cover input, relevance values are concentrated on a small number of pixels, which obtain a relevance with notably higher amplitude than that of surrounding pixels. Similarly, the large degree of variability in the relevance maps makes it difficult to identify. For the UNet, relevance values are distributed over a larger spatial domain and relevance magnitude is largest for the cloud cover field and the v-component of the wind field. Likely, this is caused by the multi-scale properties of the UNet architecture, and confirms that the UNet manages to learn features on larger spatial scales. Notably, the distribution of relevance values also displays stronger spatial correlations, which might suggests that the model learns to pay attention to spatially coherent features in the data. Also, in contrast to ResNet, the relevance of the orography field is smaller. Intuitively, this relevance attribution appears more understandable, since the selected reference pixel corresponds to a location in the mid of the Pacific Ocean, where the impact of orography on physical processes should be weak.

A prominent feature of the WB dataset is the temporal coherence of subsequent samples, which is determined by the day-night cycle, as well as the seasonal cycle. To assess the stability of the relevance maps, as well as the impact of the day-night cycle on the model mapping, we show two additional relevance maps for the UNet model applied to data samples from May 15, 2004, 12h and 20h in Figs. A3 and A4 in the appendix. The figures show that the relevance maps for 08h and 20h look very similar. In particular, clusters of spatially coherent regions of positive or negative relevance are preserved, which suggests a stability and coherence in the visual structure of the relevance maps. The maps for 12h deviate slightly and show larger relevance as for the 2 m-temperature with respect to the top-of-atmosphere incoming solar radiation, which is consistent with physical intuition.

Overall, we conclude that the different architectures learn distinct mappings, despite being trained on the same task and achieving similar prediction accuracy. Yet, we find that some aspects of the dependency structure can be partly reverse engineered via through investigation of the relevance maps. Similar statements apply to the relevance maps for models trained on the CSEns dataset. Exemplary relevance maps for this dataset are shown in Fig. B8. A more detailed analysis of the derived relevance maps, as well as the study of more specific meteorological events and weather situations at selected times and locations, is however beyond the scope of this paper and will be addressed in future work.

6. Conclusion

We have introduced an alternative way to perform deep learning-based variable-to-variable transfer. Instead of building upon the similarity of latent-space representations of parameter fields to determine transferable parameter pairs, we train a network using different transfer scenarios and select the best parameter setting. In this way, we give more flexibility to the network to exploit inter-parameter relationships, i.e., to learn parameter combinations for improved transfer. This allows saving bandwidth in in-situ settings, and can help to more aggressively compress multi-parameter simulation data. As shown in Figs. A5, B9 in the appendix, V2V transfer cannot compete with classical lossy data compression schemes in terms of compression rate, yet it may effectively support such schemes when the structure of the relevance maps generated via LRP is exploited to select spatially varying bitrates according to the importance of the data values. Other potential limitations arise due to intricacies in comparing the loss values of different parameter fields. Comparing L_1 loss values performs well on our data, which we verify by showing visualizations of the reconstructed fields. Yet it may happen that differences in the statistical distribution of field values result in deceptively low or high loss values for certain fields, which do not accurately reflect the reconstruction quality. In such cases, it may be useful to explore alternative metrics which are more robust to differences in data distributions, such as relative improvement metrics which compare against simpler baseline models such as climatologies, or metrics like SSIM (Wang et al. 2004) which relate to visualization quality. The choice of such metrics, however, may depend strongly on the dataset at hand, as well as on the intended application. For reasons of general applicability, we rely on the L_1 loss in this study, and demonstrate that it can serve as a reasonable default choice.

We have further analyzed the regional parameter structures that have the most significant effect on the reconstruction quality. By using an extension to layer-wise relevance propagation (LRP), we were able to determine regions over which the field values have a large effect on the local reconstruction accuracy. LRP results demonstrate that different model architecture learn different mapping functions, depending on the inductive bias of the used architecture. In our study, the use of the UNet model led to more physically interpretable relevance maps, while the mappings learned by the ResNet architecture are constrained in learning spatial dependencies due to the construction of the network architecture. Information like this may help in operational model applications to gain a better understanding of model-driven inference procedures and increase trustworthiness of data-driven model predictions.

In the future, we will shed light on the use of the proposed V2V approach with 3D and especially large forecast ensembles. In our current use cases, all parameter fields show rather low mutual similarities, and, thus, one can expect our approach to perform even more effective once parameter fields with certain similarities and more pronounced spatial relationships are given, like ensemble simulations. One specific task we envision is to analyse the representativeness of the single members captured by a Grand Ensemble, by using V2V to reconstruct an as small as possible subset of all members capturing the full ensemble spread. This can facilitate guidance towards weather situations that are under- or over-represented in the ensemble, and reveal situations which are intrinsically difficult to resolve. Furthermore, we intend to consider the temporal evolution of the fields to improve the reconstruction at a certain time, i.e., by letting the network train on multiple timesteps from the past.

Finally, together with meteorologists and climatologists we intend to further analyse the sensitivity maps that have been derived via LRP. Such an analysis includes the extraction of specific local weather events such as jet-cores or fronts, and to set them into relation to the regions that have been deemed important for achieving high reconstruction accuracy. A limitation of the current LRP approach lies in the necessity of selecting reference locations, for which "point-to-field" relevance maps shall be computed. In exploratory data analysis tasks, it might be non-trivial to make sensible decisions about which locations to look at in a first place. We therefore plan on refining the LRP-based analysis procedures to detect regions of high impact in an automated fashion and with a more global view to enable the interactive exploration "field-to-field" relevance relations. In a similar line of reasoning, we intend to include the time dimension in the analysis, e.g., by using temporal coherence and recurrence in the data to reduce the noise level of the derived LRP maps via temporal filtering or climatological summarization of relevances. Further efforts will be put on the investigation of alternative mechanisms for pursuing a sensitivity analysis, focusing more on spatial as well as temporal relationships between different parameters.

Acknowledgments. This study has been done within the subproject A7 “Visualization of coherence and variation in meteorological dynamics“ of the Transregional Collaborative Research Center SFB/TRR 165 “Waves to Weather“ funded by the German Research Foundation (DFG).

Data availability statement. WeatherBench dataset (Rasp et al. 2020) is publicly available at `https://github.com/pangeo-data/WeatherBench`. Access to the convective-scale ensemble data can be requested from the authors of the dataset, Necker et al. (2020). The code for the experiments is made publically available at `https://github.com/FatemehFarokhmanesh/DNN-based-Parameter-Transfer-in-Meteorological-Data.git`.

APPENDIX A

Supplementary Visualizations for the WeatherBench Dataset

The appendix provides supplementary figures illustrating specific aspects of V2V transfer in the first dataset, the WeatherBench renalysis (WB) dataset.

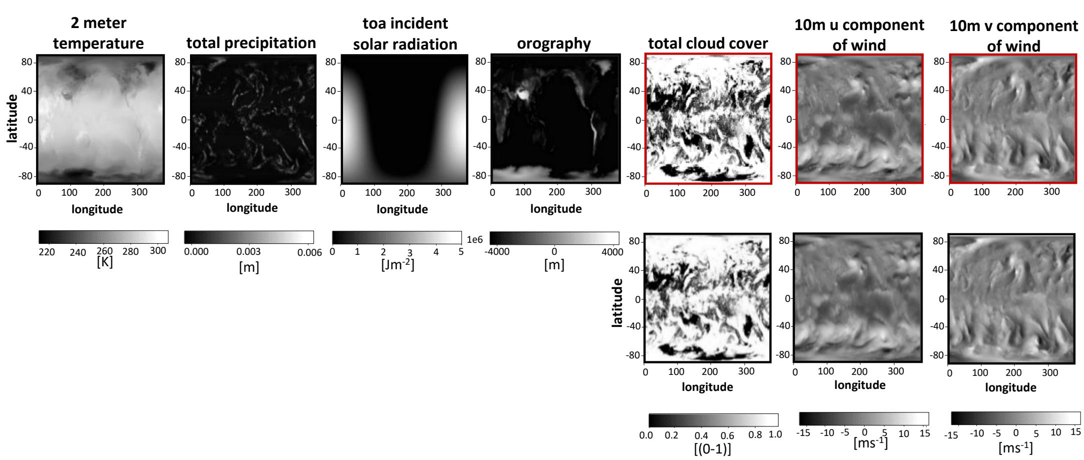


Fig. A1. Reconstruction results for the WeatherBench dataset when the network is trained to predict three parameter fields (worst combination) from four input fields. Top: The initial parameter fields. A red outline indicates those fields the network has learned to predict from the others. Bottom: Predicted parameter fields.

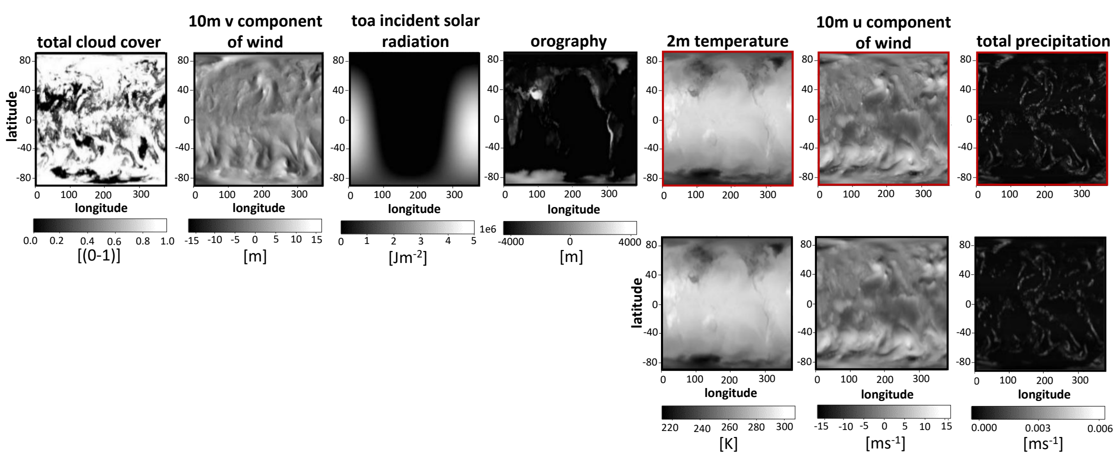


FIG. A2. Same as Fig. 7, but using UNet for training.

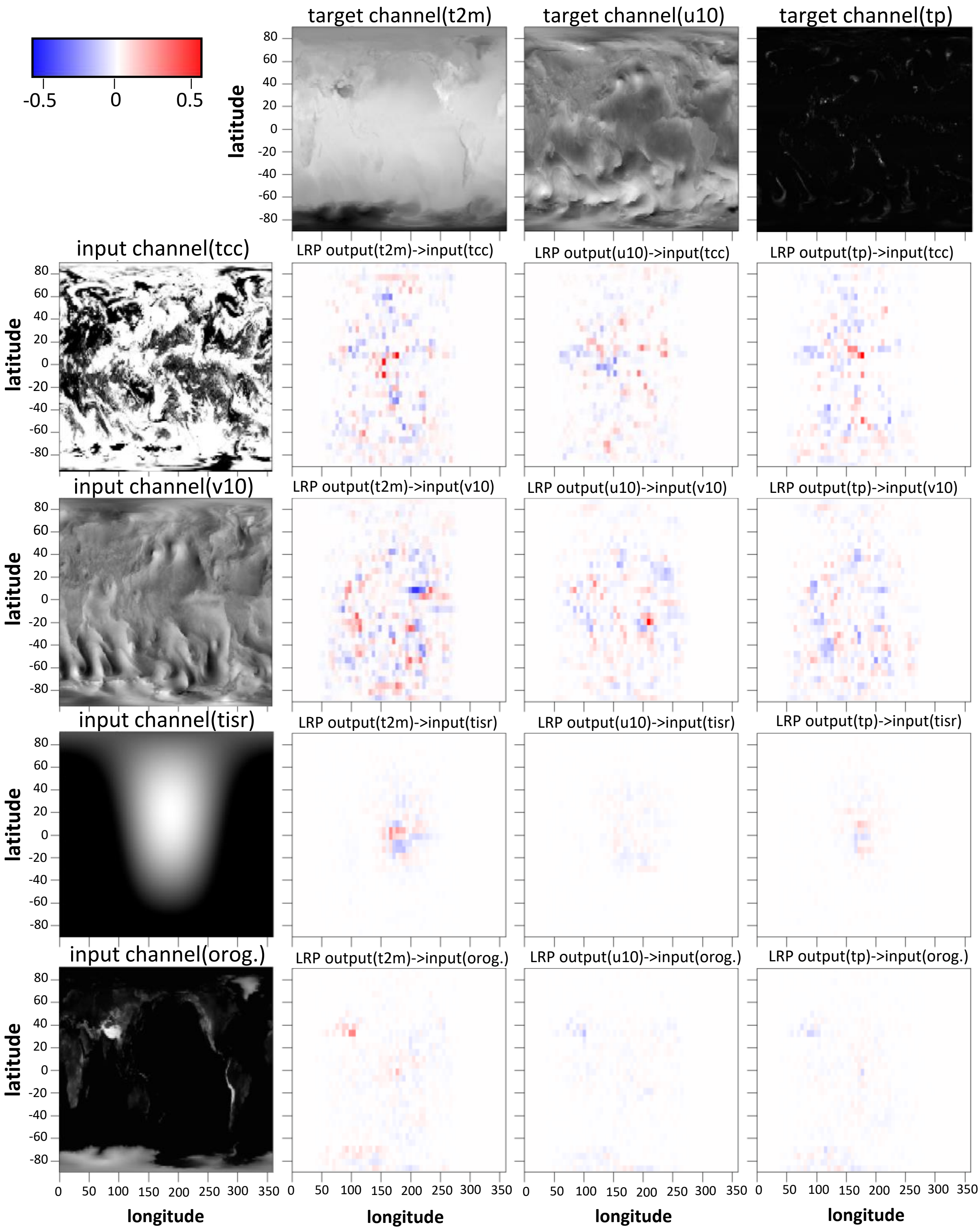
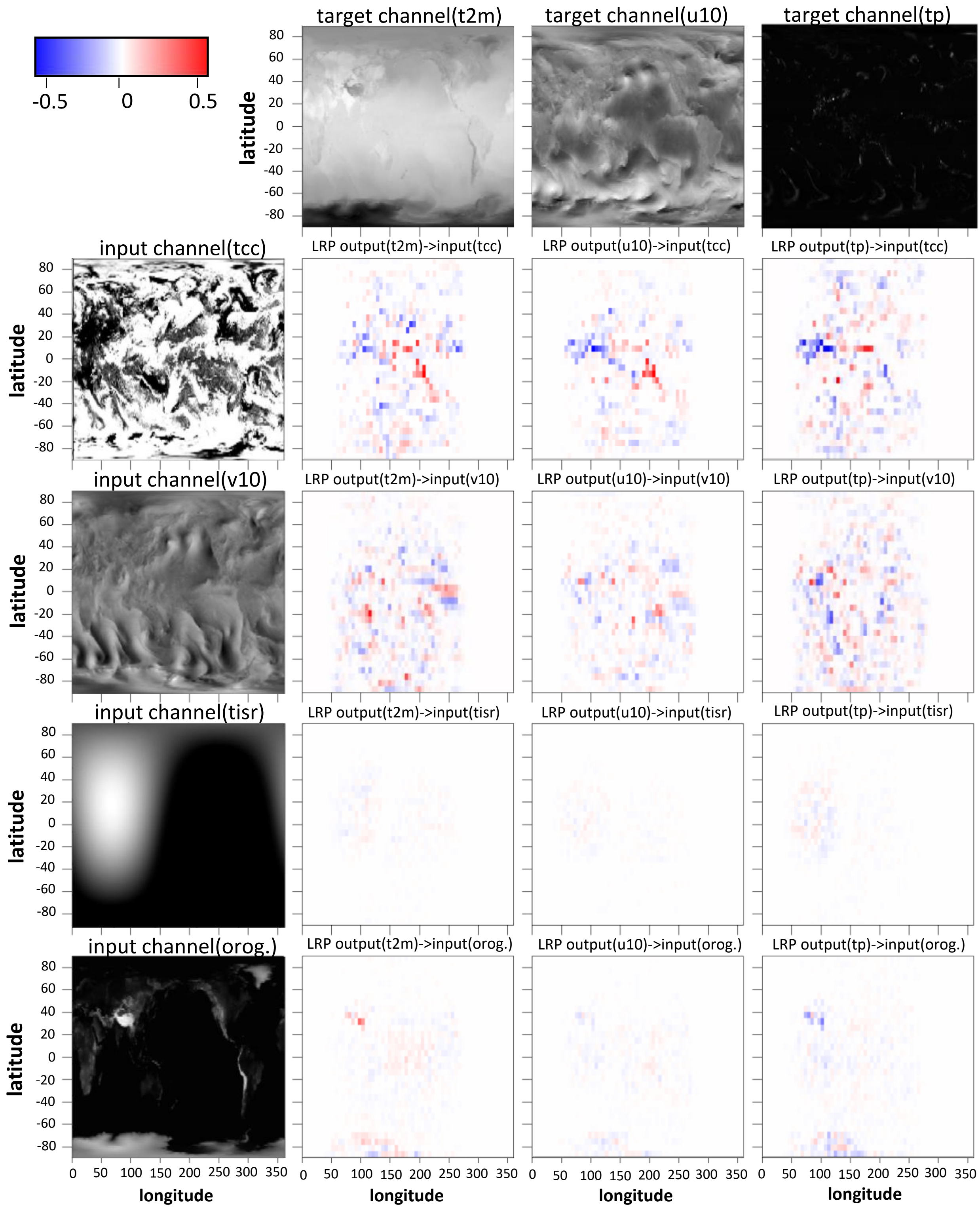
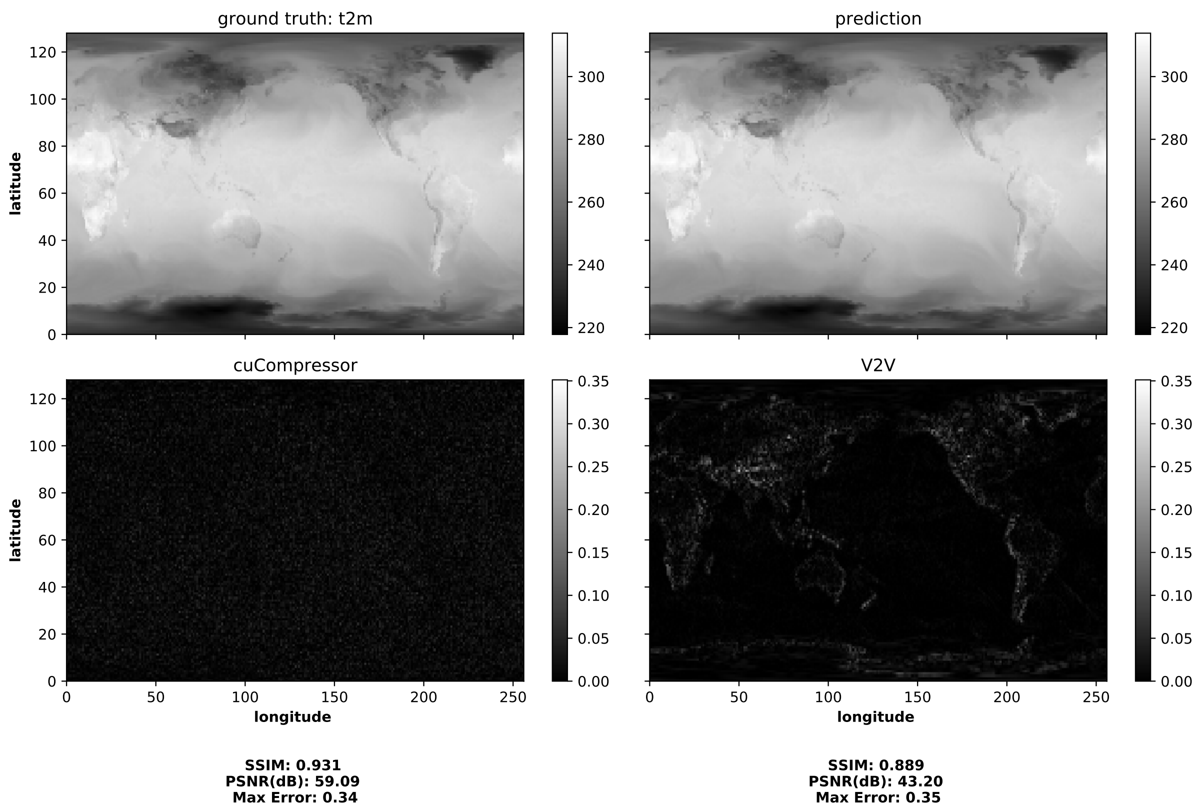


FIG. A3. LRP relevance maps with deviation-based selector function for the UNet model in the best input-output configuration for WB data. Timestamp of data sample: May 15, 2004, 12h.



629 FIG. A4. LRP relevance maps with deviation-based selector function for the UNet model in the best input-output
630 configuration for WB data. Timestamp of data sample: May 15, 2004, 20h.



631 FIG. A5. Quality comparison of V2V against a dedicated compression algorithm for volumetric data. Parameter field t2m compressed at a rate of 12:1 with the publicly available CUDA compression library by Treib et al. (2012), which provides lossy compression using a combination of the discrete wavelet transform, coefficient quantization, run-length enconding, and Huffman coding. Top left: Original field, top right: Parameter field predicted using 3-to-3 V2V transfer. Bottom: Pixel-wise differences for reconstructed compressed field and V2V reconstruction.

APPENDIX B

Variable-to-Variable Transfer for the Convective-Scale Ensemble

The convective-scale ensemble simulation (CSEns), generated by Necker et al. (2020), contains 1000 runs of a 3D atmospheric dynamics model over a rectangular domain in central Europe. Data are stored on a regular grid with 352×250 nodes, which corresponds to a horizontal grid spacing of 3 km and allows the resolution of convective effects in the model dynamics. The simulation covers a time interval of six hours, with a period of one hour between successive time steps, and comprises 30 levels in height. In the lower levels, some of the data are invalid due to grid cells falling below the level of the surface topography. Levels with missing values are omitted. 3D data are available for a total of 9 different parameters, which are temperature (tk), u-, v-, and w- component of winds (u, v, w), geopotential height (z), relative humidity (rh), mixing ratio of all hydro meteors (qh), water vapor mixing ratio (qv), and radar reflectivity (dbz). Structural differences are observed not only between different parameters, but also between different timesteps and height levels of the same parameter. Specifically, the variation in the fields decreases with increasing distance from the earth surface, due to decreasing influence of boundary layer effects, and complexity increases with increasing simulation time due to a strengthening of convective activity. To enable a fair comparison between the CSEns and WB, we consider data only for the three lowermost levels without missing values, as well as the three latest time steps, which show the highest field complexity. We further split time-variate 3D fields both in time and height to obtain a sequence of plain 2D fields. We then consider data for 200 members for training, five members for validation, and the remaining members for testing and visualization.

The appendix provides additional figures illustrating V2V transfer in the second dataset, the convective-scale ensemble (CSEns) by Necker et al. (2020), which were excluded from the main paper to improve readability.

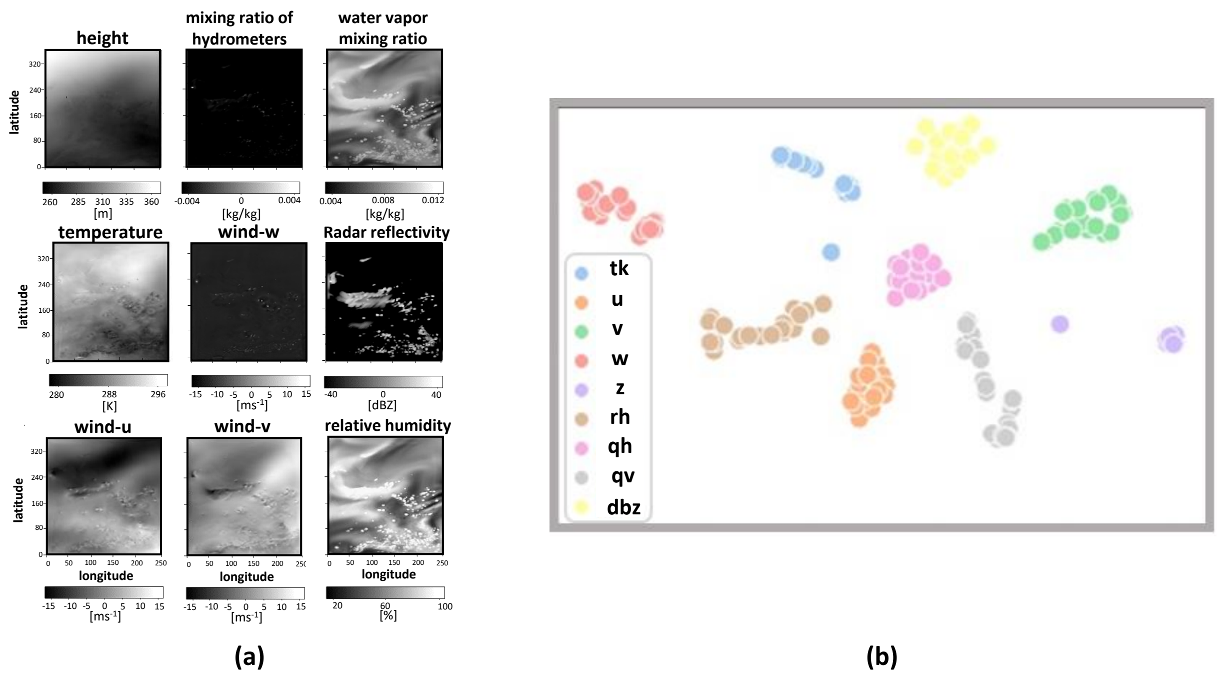


Fig. B1. Different parameter fields in the CSEns dataset. a) Gray-scale visualizations of the parameter fields at a particular time. b) t-SNE projections of latent-space features of the parameter fields (different parameters indicated by colors) at different times (note that projections for different initializations of t-SNE yield similar groupings).

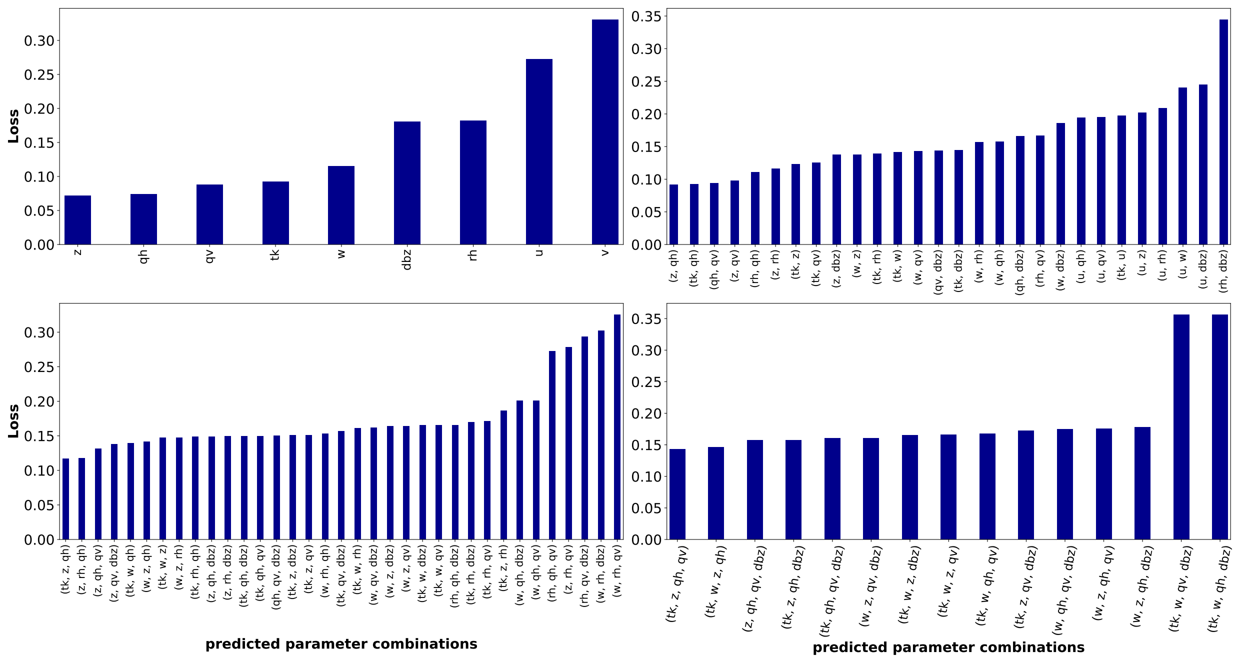


Fig. B2. Bar charts showing the losses of all networks trained for 4-to-5 parameter transfer with the CSEns dataset using the proposed iterative loss-based approach. Top left: first iteration, top right: second iteration, bottom left: third iteration, bottom right: fourth iteration. Bars represent losses after five epochs of training.

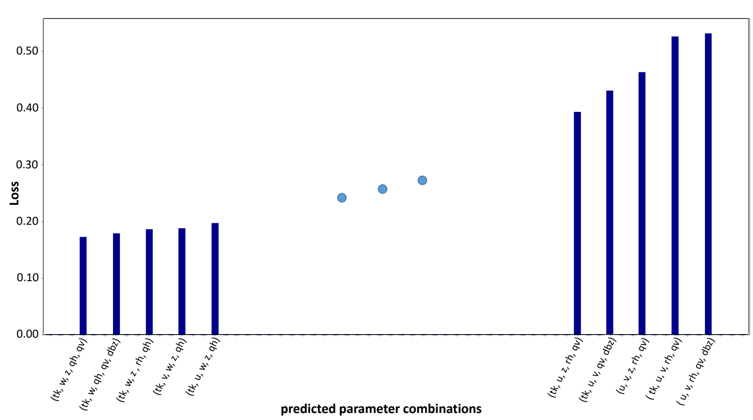


Fig. B3. Bar chart showing the losses of the best (left) and worst (right) possible networks for 4-to-5 parameter transfer with the CSEns dataset. All $\binom{9}{5}$ possible models have been trained for five epochs. Configurations with intermediate losses have been omitted from the chart for clarity of the visualization.

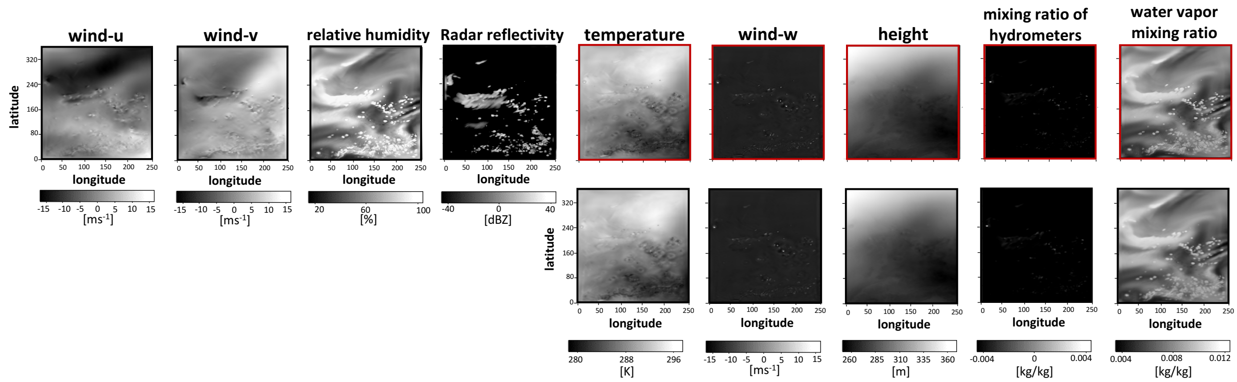


FIG. B4. Reconstruction results for the CSEns datasets when the network is trained to predict five parameter fields from four input fields. Top: The initial parameter fields. A red outline indicates those fields the network has learned to predict from the others. Bottom: Predicted parameter fields.

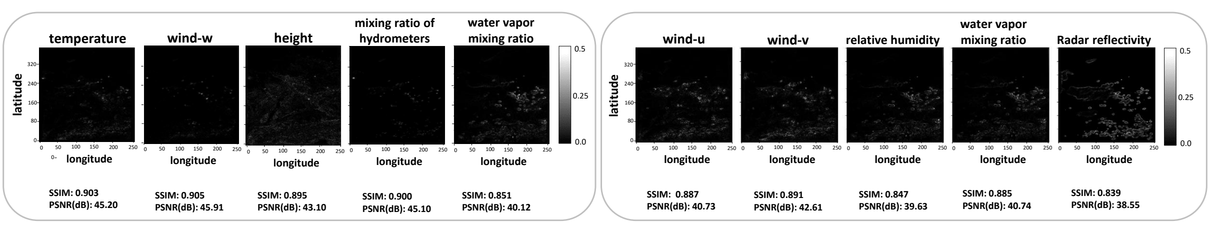


FIG. B5. Same as Fig. 8, but using the CSEns dataset.

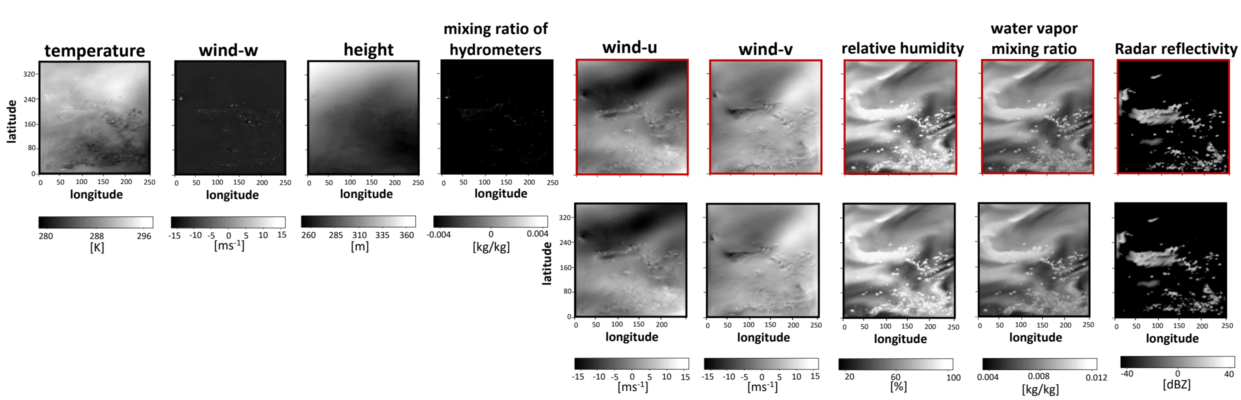


FIG. B6. Same as Fig. A1, but using the CSEns dataset.

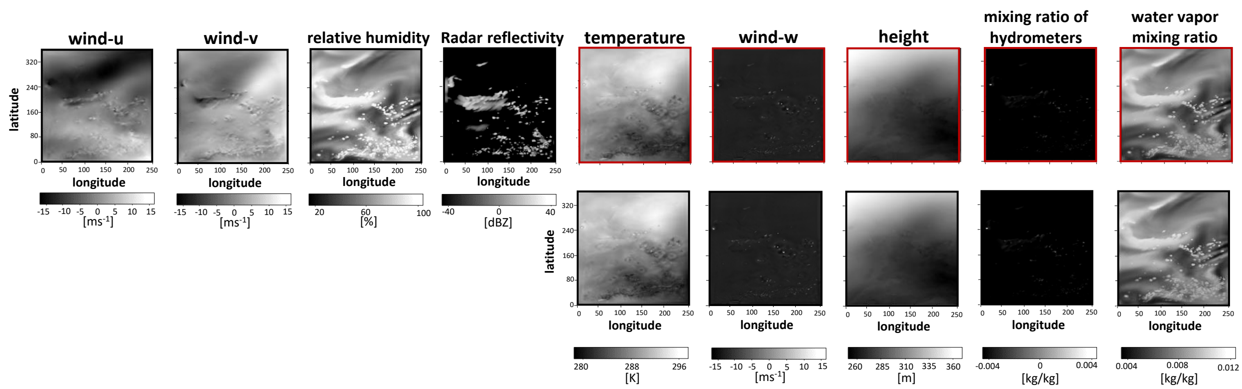
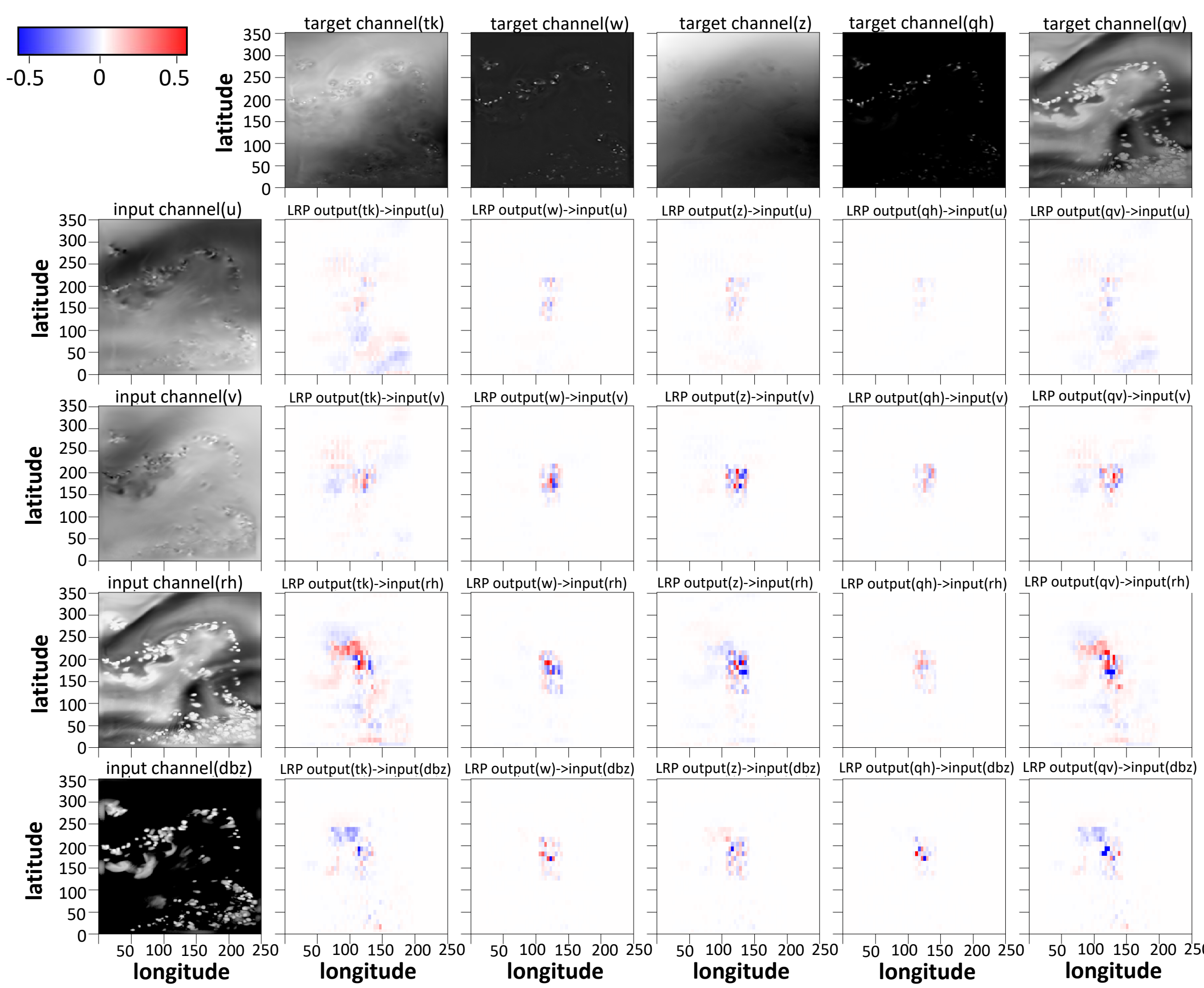


FIG. B7. Same as Fig. B4, but using the UNet architecture instead of the ResNet.



674 FIG. B8. LRP relevance maps with deviation-based selector function for the UNet model in the best input-output configuration for CSEns data. Timestamp of data sample: June 1, 2016, 17h.

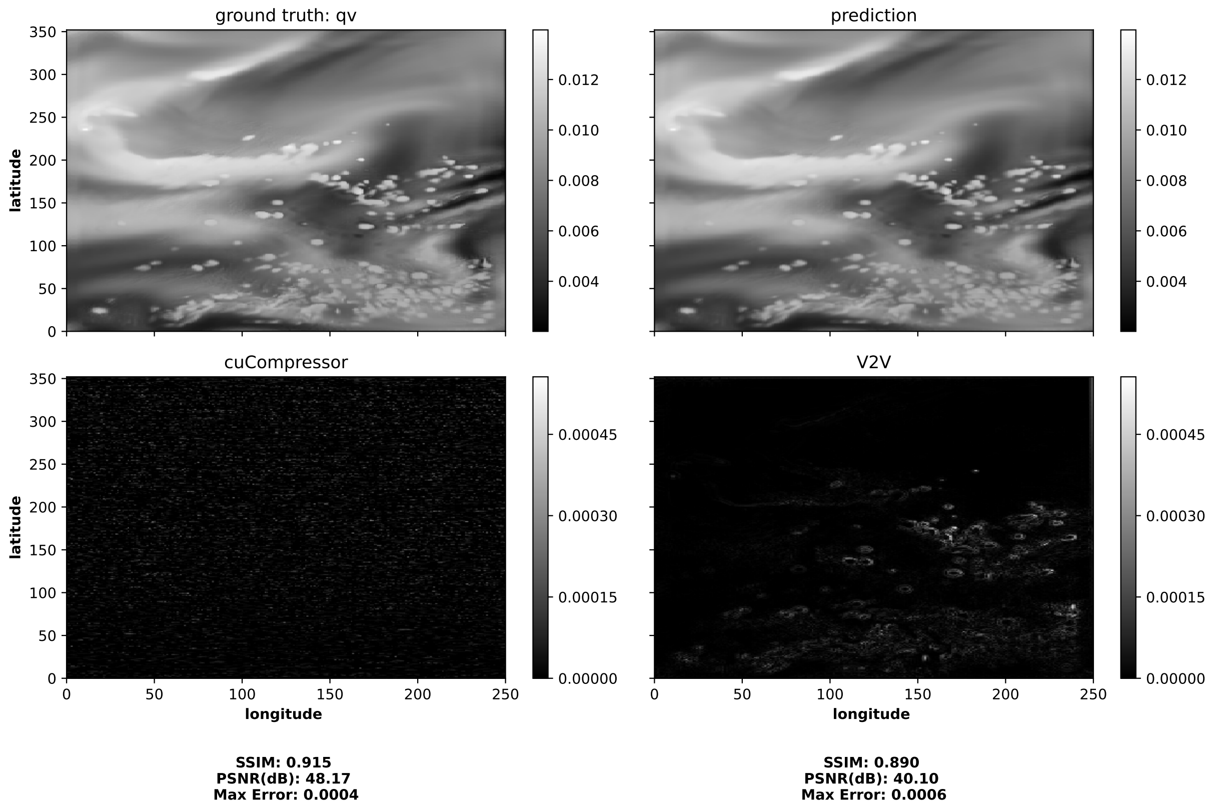


FIG. B9. Quality comparison of V2V against a dedicated compression algorithm for volumetric data. Parameter field qv compressed at a rate of 12:1 with the publicly available CUDA compression library by Treib et al. (2012), which provides lossy compression using a combination of the discrete wavelet transform, coefficient quantization, run-length enconding, and Huffman coding. Top left: Original field, top right: Parameter field predicted using 4-to-5 V2V transfer. Bottom: Pixel-wise differences for reconstructed compressed field and V2V reconstruction.

References

Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, 2015: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, **10 (7)**, e0130 140.

Cao, W., Z. Yan, Z. He, and Z. He, 2020: A comprehensive survey on geometric deep learning. *IEEE Access*, **8**, 35 929–35 949.

Cheng, J., Q. Kuang, C. Shen, J. Liu, X. Tan, and W. Liu, 2020: Reslap: Generating high-resolution climate prediction through image super-resolution. *IEEE Access*, **8**, 39 623–39 634.

Glorot, X., and Y. Bengio, 2010: Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, 249–256.

Guo, L., S. Ye, J. Han, H. Zheng, H. Gao, D. Z. Chen, J.-X. Wang, and C. Wang, 2020: SSR-VFD: Spatial super-resolution for vector field data analysis and visualization. *2020 IEEE Pacific Visualization Symposium (PacificVis)*, IEEE Computer Society, 71–80.

Han, J., and C. Wang, 2019: TSR-TVD: Temporal super-resolution for time-varying data analysis and visualization. *IEEE transactions on visualization and computer graphics*, **26 (1)**, 205–215.

Han, J., and C. Wang, 2020: SSR-TVD: Spatial super-resolution for time-varying data analysis and visualization. *IEEE Transactions on Visualization and Computer Graphics*.

Han, J., H. Zheng, D. Z. Chen, and C. Wang, 2021a: STNet: An end-to-end generative framework for synthesizing spatiotemporal super-resolution volumes. *IEEE Transactions on Visualization and Computer Graphics*, 1–1.

Han, J., H. Zheng, Y. Xing, D. Z. Chen, and C. Wang, 2021b: V2v: A deep learning approach to variable-to-variable selection and translation for multivariate time-varying data. *IEEE Transactions on Visualization and Computer Graphics*, **27**, 1290–1300.

He, K., X. Zhang, S. Ren, and J. Sun, 2016: Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Helbig, C., L. Bilke, H.-S. Bauer, M. Böttinger, and O. Kolditz, 2015: Meva-an interactive visualization application for validation of multifaceted meteorological data with multiple 3d devices. *PloS one*, **10 (4)**, e0123 811.

Hersbach, H., and Coauthors, 2020: The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, **146 (730)**, 1999–2049.

Höhlein, K., M. Kern, T. Hewson, and R. Westermann, 2020: A comparative study of convolutional neural network models for wind field downscaling. *Meteorological Applications*, **27 (6)**, https://doi.org/https://doi.org/10.1002/met.1961.

Ioffe, S., and C. Szegedy, 2015: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning*, PMLR, 448–456.

Kokhlikyan, N., and Coauthors, 2020: Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*.

Necker, T., S. Geiss, M. Weissmann, J. Ruiz, T. Miyoshi, and G.-Y. Lien, 2020: A convective-scale 1,000-member ensemble simulation and potential applications. *Quarterly Journal of the Royal Meteorological Society*, **146 (728)**, 1423–1442.

Pouliot, D., R. Latifovic, J. Pasher, and J. Duffe, 2018: Landsat super-resolution enhancement using convolution neural networks and sentinel-2 for training. *Remote Sensing*, **10 (3)**, 394.

Rasp, S., P. D. Dueben, S. Scher, J. A. Weyn, S. Mouatadid, and N. Thuerey, 2020: Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, **12 (11)**, e2020MS002 203.

Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Coauthors, 2019: Deep learning and process understanding for data-driven earth system science. *Nature*, **566 (7743)**, 195–204.

Röber, N., and J. F. Engels, 2019: In-situ processing in climate science. *International Conference on High Performance Computing*, Springer, 612–622.

Rodrigues, E. R., I. Oliveira, R. Cunha, and M. Netto, 2018: Deepdownscale: a deep learning strategy for high-resolution weather forecast. *2018 IEEE 14th International Conference on e-Science (e-Science)*, IEEE, 415–422.

Ronneberger, O., P. Fischer, and T. Brox, 2015: U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.

Sato, T., O. Tatebe, and H. Kusaka, 2019: In-situ data analysis system for high resolution meteorological large eddy simulation model. *Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, 155–158.

Serifi, A., T. Günther, and N. Ban, 2021: Spatio-temporal downscaling of climate data using convolutional and error-predicting neural networks. *Frontiers in Climate*, **3**, https://doi.org/10.3389/fclim.2021.656479.

Toderici, G., D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, 2017: Full resolution image compression with recurrent neural networks. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 5306–5314.

Treib, M., F. Reichl, S. Auer, and R. Westermann, 2012: Interactive editing of gigasample terrain fields. *Computer Graphics Forum (Proc. Eurographics)*, **31 (2)**, 383–392, https://doi.org/10.1111/j.1467-8659.2012.03017.x, URL http://diglib.eg.org/EG/CGF/volume31/issue2/v31i2pp383-392.pdf.

van der Maaten, L., and G. Hinton, 2008: Visualizing data using t-SNE. *Journal of Machine Learning Research*, **9**, 2579–2605, URL http://www.jmlr.org/papers/v9/vandermaaten08a.html.

Wang, C., H. Yu, R. W. Grout, K.-L. Ma, and J. H. Chen, 2011: Analyzing information transfer in time-varying multivariate data. *2011 IEEE Pacific Visualization Symposium*, 99–106, https://doi.org/10.1109/PACIFICVIS.2011.5742378.

Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, 2004: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, **13 (4)**, 600–612.

Weiss, S., M. Chu, N. Thuerey, and R. Westermann, 2019: Volumetric isosurface rendering with deep learning-based super-resolution. *IEEE Transactions on Visualization and Computer Graphics*, 1–1.

Weiss, S., M. Işık, J. Thies, and R. Westermann, 2020: Learning adaptive sampling and reconstruction for volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 1–1.

Zhou, Z., Y. Hou, Q. Wang, G. Chen, J. Lu, Y. Tao, and H. Lin, 2017: Volume upscaling with convolutional neural networks. *Proceedings of the Computer Graphics International Conference*, 1–6.