Journal of Machine Learning Research 24 (2023) 1-40 Submitted 2/23; Revised 7/23; Published 12/23

Hard-Constrained Deep Learning for Climate Downscaling

Paula Harder PAULA.HARDER@MILA.QUEBEC
Fraunhofer ITWM, Kaiserslautern, Germany
Mila Quebec AI Institute, Montreal, Canada

Alex Hernandez-Garcia*
Mila Quebec AI Institute, Montreal, Canada
University of Montreal, Montreal, Canada

Venkatesh Ramesh*
Mila Quebec AI Institute, Montreal, Canada
University of Montreal, Montreal, Canada

Qidong Yang
Mila Quebec AI Institute, Montreal, Canada
New York University, New York, USA

Prasanna Sattegeri
IBM Research, New York, USA

Daniela Szwarcman
IBM Research, Brazil

Campbell D. Watson
IBM Research, New York, USA

David Rolnick
Mila Quebec AI Institute, Montreal, Canada
McGill University, Montreal, Canada

Editor: Shakir Mohamed

Abstract

The availability of reliable, high-resolution climate and weather data is important to inform long-term decisions on climate adaptation and mitigation and to guide rapid responses to extreme events. Forecasting models are limited by computational costs and, therefore, often generate coarse-resolution predictions. Statistical downscaling, including super-resolution methods from deep learning, can provide an efficient method of upsampling low-resolution data. However, despite achieving visually compelling results in some cases, such models frequently violate conservation laws when predicting physical variables. In order to conserve physical quantities, here we introduce methods that guarantee statistical constraints are satisfied by a deep learning downscaling model, while also improving their performance according to traditional metrics. We compare different constraining approaches and demonstrate their applicability across different neural architectures as well as a variety of climate and weather data sets. Besides enabling faster and more accurate climate predictions through downscaling, we also show that our novel methodologies can improve super-resolution for satellite data and natural images data sets.

∗. Equal contribution.

1. Introduction

Accurate modeling of weather and climate is critical for taking effective action to combat climate change. In addition to shaping global understanding of climate change, local and regional predictions guide adaptation decisions and provide the impetus for action to reduce greenhouse gas emissions (Gutowski et al., 2020). Predicted and observed quantities such as precipitation, wind speed, and temperature impact decisions in sectors such as agriculture, energy, and transportation. While these quantities are often required at a fine geographical and temporal scale to ensure informed decision-making, most climate and weather models are extremely computationally expensive to run (sometimes taking months even on supercomputers), resulting in coarse-resolution predictions. Thus, there is a need for fast methods that can generate high-resolution data based on the low-resolution models that are commonly available.

The terms *downscaling* in climate science and *super-resolution* (SR) in machine learning (ML) refer to a map from low-resolution (LR) input data to high-resolution (HR) versions of that same data; the high-resolution output is referred to as the super-resolved (SR) data. Downscaling via established statistical methods—*statistical downscaling*—has been long used by the climate science community to increase the resolution of climate data (Maraun and Widmann, 2018). In statistical downscaling, there are two subfields, *perfect prognosis* and *model output statistics* (Maraun and Widmann, 2018). Whereas perfect prognosis learns the relationship between LR and HR observations, model output statistics learns directly the function from model output to observations, including a form of bias correction.

In perfect prognosis, predictands and predictors usually include different variables. If both inputs and outputs consist of the same variables, this is referred to as super-resolution, even in a climate context. In parallel, computer vision SR has evolved rapidly using various deep learning architectures, with such methods now including super-resolution convolutional neural networks (CNNs) (Dong et al., 2016), generative adversarial models (GANs) (Wang et al., 2018a), vision transformers (Yang et al., 2020), and normalizing flows (Lugmayr et al., 2020). Increasing the temporal resolution via frame interpolation is also an active area of research for video enhancement (Liu et al., 2017) that can be transferred to spatiotemporal climate data. Recently, deep learning approaches have been applied to a variety of climate and weather data sets, covering both model output data and observations. In addition to using neural networks to learn parametrization, replace model parts in a hybrid setup, or run full forecasts, downscaling is a field for deep learning to improve and accelerate Earth system simulations (Reichstein et al., 2019). Climate super-resolution has mostly focused on CNNs (Vandal et al., 2017), recently shifting toward GANs (Stengel et al., 2020; Wang et al., 2021).

Most statistical downscaling tools are applied offline as a tool for post-processing. In that case, machine learning methods can be directly employed on the output data, following data reformatting. However, downscaling tools could be applied online within a global climate model too (e.g. Quiquet et al. (2018)), where a lower resolution output of a climate model part is downscaled, and its high-resolution version is fed back into the climate model.

There are certain tasks that are more suited for hard-constraining than others. One important point is that there exists a relationship between low-resolution and high-resolution samples for downscaling or between input and output for other tasks, given by an equation. This can be the case when modeling physical quantities, with, for example, mass or energy

conservation that exists between LR and HR pairs. On the one hand, if we consider compressed or blurry images and the task is to remove the effects of compression or blur, there may be no known constraint between low and high resolution, so constraining methodologies would not be applicable. On the other hand, for some data from e.g. satellites or telescopes, images are created by summing photons across a given field of view, so the value at a given pixel can be interpreted as the sum of values at unobserved subpixels; in such cases, hard constraints could potentially be useful.

In this work, we introduce novel methods to strictly enforce physics-inspired consistency constraints between low-resolution (input) and high-resolution (output) images. We do this via a constraint layer at the end of a neural architecture, which renormalizes the prediction either additively, multiplicatively, or with an adaptation of the softmax layer. We use climate and weather data sets based on European Center for Medium-Range Weather Forecasts (ECMWF) reanalysis data version 5 (ERA5) (Hersbach et al., 2020), Weather Research and Forecast Model (WRF) data (Auger et al., 2021), and the Norwegian Earth System Model (NorESM) (Seland et al., 2020) data, spanning different quantities such as water content, temperature, water vapor, and liquid water content. For the ERA5 data, we increase the resolution by different factors, we create data sets with an enhancement of factors ranging from 2 over 4 and 8 to 16. We show the utility of our methods across architectures including CNNs, GANs, CNN-RNNs, and a novel architecture that we introduce to apply super-resolution in both spatial and temporal dimensions. Besides climate data sets, we show that our methods are able to improve predictive accuracies for lunar satellite imagery super-resolution as well as on standard image super-resolution benchmark data sets, like Set5, Set14, Urban100 and BSD100. Our code is available at `https://github.com/RolnickLab/constrained-downscaling` and our main data set can be found at `https://drive.google.com/file/d/1IENhP1-aTYyqOkRcnmCIvxXkvUW2Qbdx/view`.

Contributions Our main contributions can be summarized as follows:

- We introduce a novel constraining methodology for deep learning-based downscaling methods, which guarantees that physical consistency constraints such as mass and energy conservation between low-resolution and high-resolution are satisfied.
- We show that our method improves predictive performance across different deep learning architectures on a variety of climate data sets.
- Additionally, we show that our method increases the accuracy of super-resolution in other domains, such as natural images and satellite imagery.
- Finally, we introduce a new deep learning architecture for downscaling along both spatial and temporal dimensions.

2. Related work

Deep Learning for Climate Downscaling There exists extensive work on ML methods for climate and weather observation and prediction downscaling, from CNN architectures (Vandal et al., 2017) to GANs (Stengel et al., 2020) and normalizing flows (Groenke et al., 2020). Recently, GANs have become a very popular architecture choice, including many

works on precipitation model downscaling (Wang et al., 2021; Watson et al., 2020; Chaudhuri and Robertson, 2020) as well as other quantities such as wind and solar data (Stengel et al., 2020). Unified frameworks comparing methods and benchmarks were introduced by Baño Medina et al. (2020) to assess different SR-CNN setups and by Kurinchi-Vendhan et al. (2021) with the introduction of a new data set for wind and solar SR. To date, there has been limited work on spatiotemporal SR with climate data. Some authors have looked at super-resolving multiple time steps at once without increasing the temporal resolution (Harilal et al., 2021; Leinonen et al., 2021). Serifi et al. (2021) did increase the temporal resolution by simply treating the time steps as different channels and using a standard SR-CNN.

Constrained Learning for Climate Various works on ML for climate science have attempted to enforce certain physical constraints via soft penalties in the loss (Beucler et al., 2019), linearly constrained neural networks for convection (Beucler et al., 2021), or aerosol microphysics emulation (Harder et al., 2022) using completion or correction methods. Zanna and Bolton (2020) and Zanna and Bolton (2021) use a final fixed convolutional layer to achieve momentum and vorticity conservation in an ML ocean model. A different line of work incorporates constraints into machine learning based on flux balances (Sturm and Wexler, 2020, 2022; Yuval et al., 2021). These strategies use domain knowledge of how properties flow to ensure conservation of different quantities. Instead of predicting tendencies directly, fluxes are predicted. Hess et al. (2022) introduces one global constraint to be applied to bias-correct the precipitation prediction generated by a GAN. Outside of climate science, recent work has emerged on enforcing hard constraints on the output of neural networks (e.g. Donti et al. (2021)).

Constrained Learning for Downscaling In super-resolution for turbulent flows, MeshfreeFlowNet (Jiang et al., 2020) employs a physics-informed model which adds PDEs as regularization terms to the loss function. In parallel to our work, the first approaches employing hard constraints for climate-related downscaling were introduced: Geiss and Hardin (2023) introduced an enforcement operator applied to multiple CNN architectures for scientific data sets. A CNN with a multiplicative renormalization layer is used for atmospheric chemistry model downscaling in Geiss et al. (2022). We are the first to compare a variety of different hard-constraining approaches and also apply them to multiple deep learning architectures.

3. Enforcing constraints

When modeling physical quantities such as precipitation or water mass, principled relationships such as mass conservation can naturally be established between low-resolution and high-resolution samples. Here, we introduce a new methodology to incorporate these constraints within a neural network architecture. We choose hard constraints enforced through the architecture over soft constraints that use an additional loss term. Hard constraints guarantee certain constraints even at inference time, whereas soft constraining encourages the network to output values that are close to satisfying constraints, by minimizing a penalty during training, but do not provide any guarantees. Additionally, for our case hard constraining increases the predictive ability, and soft constraining can lead to unstable training and an accuracy-constraints trade-off (Harder et al., 2022). Adding hard constraints restricts the hypothesis space to a smaller subspace that satisfies the constraints. With that, we

reformulate the learning problem to an easier problem and achieve better results including prior knowledge.

3.1 Setup

Consider the case of downscaling low-resolution pixels x by a factor of N in each linear dimension, and let $n := N^2$. Let $y_i, i = 1, \ldots, n$ be the values in the predicted high-resolution patch that correspond to x. The set $\{y_i\}$ for $i = 1, \ldots, n$ is also referred to as a super-pixel. Then, a conservation law takes the form of the following constraint:

$$\frac{1}{n}\sum_{i=1}^{n} y_i = x. \tag{1}$$

Depending on the predicted quantity, there may additionally be an inequality constraint associated with the data. In our work, there was only one example, concerning the positivity of several physical quantities (e.g. water mass). The inequality for this case would be:

$$\forall i \in [[1, n]], y_i \geq 0. \tag{2}$$

We note that the methodologies we suggest in this work only deal with this special case.

3.2 Constraint layer

We introduce three different alternatives as constraint layers: additive constraining, multiplicative constraining, and softmax-based constraining. These are all added at the end of any neural architecture, as shown in Figure 2, and all satisfy Eq. 1 by construction. The constraints are applied for each pair of input pixel x and the corresponding SR $N \times N$ patch. An illustration is shown in Figure 1. We will use $\tilde{y}_i, i = 1, \ldots, n$ to denote the intermediate outputs of the neural network before the constraint layer and $y_i, i = 1, \ldots, n$ to be the final outputs after applying the constraints.

Additive constraining For our Additive Constraint Layer (AddCL), we take the intermediate outputs and reset them using the following operation:

$$y_j = \tilde{y}_j + x - \frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i. \tag{3}$$

We also consider a more complex additive approach, the Scaled Additive Constraint Layer (ScAddCL), which was introduced in parallel work to ours by Geiss and Hardin (2023):

$$y_j = \tilde{y}_j + (x - \frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i) \cdot \frac{\sigma + \tilde{y}_i}{\sigma + \frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i}, \tag{4}$$

with $\sigma := \text{sign}(\frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i - x)$, so $\sigma \in \{-1, 1\}$ The pixel values are assumed to in $[-1, 1]$. For more details see Geiss and Hardin (2023).

Multiplicative constraining For the Multiplicative Constraint Layer (MultCL) approach, we rescale the intermediate output using the corresponding input value x:

$$y_j = \tilde{y}_j \cdot \frac{x}{\frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i}. \tag{5}$$

A similar approach is used in Geiss et al. (2022). Note that this approach can violate non-negativity constraints (e.g. 18 pixels per 128x128 patch for 8× upsampling, see Table 5), so it is sometimes detrimental. Multiplicative constraining can however be generalized by introducing any function g:

$$y_j = g(\tilde{y}_j) \cdot \frac{x}{\frac{1}{n}\sum_{i=1}^{n} g(\tilde{y}_i)}. \tag{6}$$

If g is positive, the output is guaranteed to be positive too.

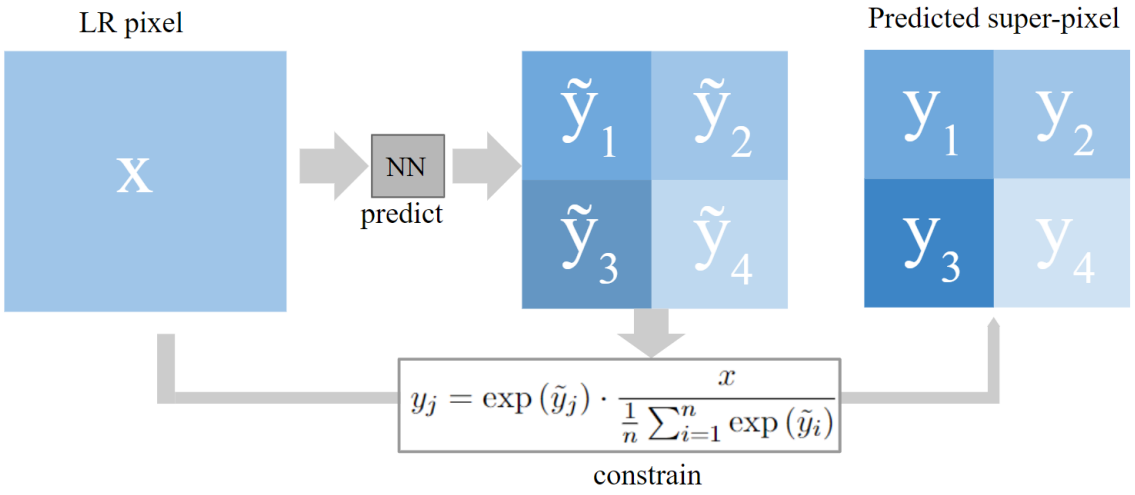

Figure 1: Our Softmax Constraining Layer (SmCL) is shown for one input pixel x and the corresponding predicted 2×2 super-pixel for the case of 2× upsampling. This layer is added at the end of a NN and enforces given constraints guaranteed by construction. Besides equality constraints, it enforces positivity of the outputs.

Softmax constraining For predicting quantities like atmospheric water content, we want to enforce the output to be non-negative for it to be physically valid. Here, we use a softmax multiplied by the corresponding input pixel value x:

$$y_j = \exp(\tilde{y}_j) \cdot \frac{x}{\frac{1}{n}\sum_{i=1}^{n} \exp(\tilde{y}_i)}. \tag{7}$$

This Softmax Constraint Layer (SmCL) is a special case of Eq. (6) with $g \equiv \exp$ and enforces $y_i \geq 0, i = 1, \ldots, n$.

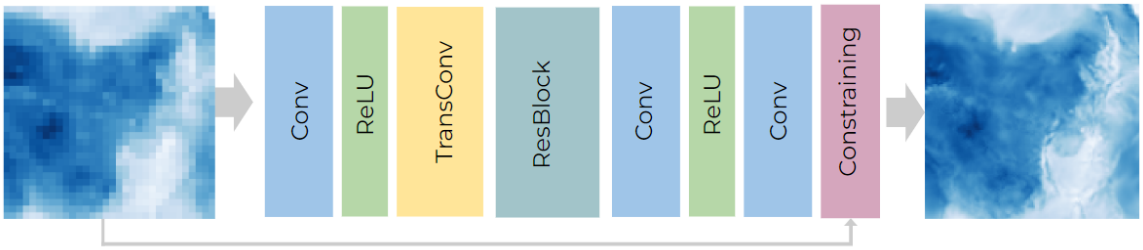

Figure 2: The CNN architecture used here for 2× upsampling including the constraint layer (in red). The LR input is passed to the last layer, the constraint layer, to enforce the constraint and produce a consistent HR output.

Differences of Constraint Layers The four different constraint layers have in common that they all enforce Eq. (1) by construction and we will see in Section 6 that the differences in performance are rather small. To point out and summarize the differences: Whereas ScAddCl ($[-1, 1]$) and MultCL (non-zero) are restricted in the range of input values they can handle, AddCL and SmCL work with any inputs. SmCL gives only positive outputs, which can be either beneficial by serving as an additional physical constraint or too restrictive if the output domain includes negative values. MultCL might get unstable for values close to zero. Additionally, the choice of constraint layer influences the variance among super-pixels, with SmCL having the highest variance (see Table 13):

3.3 Generalization of our constraining methodologies

The focus of this work is on a consistency constraint for downscaling, but the methodology is not limited to this and can be applied to different setups. It can be slightly adapted to e.g. enforce a weighted formulation of Eq. (1), global constraint, or mass conservation constraints for emulation. Here we show how our constraint layers can be employed for different cases, starting with a more general setup and then formulating special relevant cases.

3.3.1 Generalization setup

We consider the learning task (supervised or unsupervised), where $X \in \mathbb{R}^{n_{\text{in}}}$ is our input and $y \in \mathbb{R}^{n_{\text{out}}}$ the final output. Let $(I_j)_{j=1,\dots,n_p}$ be a partition of $\{1, \dots, n_{\text{out}}\}$ into n_p subsets (n_p determines how many different constraints are imposed, e.g. n_{in} for our downscaling setup), $g_{ij} : \mathcal{D} \subset \mathbb{R} \to \mathbb{R}$, $i \in I_j$ an invertible function and $h_j : \mathbb{R}^{n_{\text{out}}} \to \mathbb{R}$ an arbitrary function. The set of constraints is given by

$$\sum_{i \in I_j} g_{ij}(y_i) = h_j(X), \tag{8}$$

for each $j = 1, \dots, n_p$.

These constraints can then be enforced with the above-introduced layers restated as follows

$$\begin{aligned} y_i^{\text{AddCL}} &= g_{ij}^{-1}(\tilde{y}_i + \frac{1}{n}h_j(X) - \frac{1}{n}\sum_{k\in I_j}\tilde{y}_k), \\ y_i^{\text{MultCL}} &= g_{ij}^{-1}(\tilde{y}_i \cdot \frac{h_j(X)}{\sum_{k\in I_j}\tilde{y}_k}), \\ y_i^{\text{SmCL}} &= g_{ij}^{-1}(\exp(\tilde{y}_i) \cdot \frac{h_j(X)}{\sum_{k\in I_j}\exp(\tilde{y}_k)}), \end{aligned}$$

for $i \in I_j$ and $j = 1, \ldots, n_p$.

The main case considered in this work (Eq. (1) is a special case with $h_j(X) = nX_j$ for j indexing all super-pixels and g being the identity function. Note that MultCl and SmCL cannot be directly applied if $h_j \equiv 0$ for any j, leading to a constant prediction.

3.3.2 Weighted formulation

In an Earth system modeling context data often originates from a latitude-longitude grid. This implies that the areas in each field are not exactly the same. The downscaling consistency constraint (Eq. (1)) is then changed to a weighted formulation:

$$\frac{1}{n}\sum_{i=1}^{n}\alpha_i y_i = x. \tag{9}$$

Analogously, the AddCL, MultCl, and SmCL are reformulated as

$$\begin{aligned} y_i^{\text{AddCL}} &= \frac{1}{\alpha_i}(\tilde{y}_i + x - \frac{1}{n}\sum_{i=k}^{n}\tilde{y}_k). \\ y_i^{\text{MultCL}} &= \tilde{y}_i \cdot \frac{x}{\alpha_i \frac{1}{n}\sum_{k=1}^{n}\tilde{y}_k} \\ y_i^{\text{SmCL}} &= \exp(\tilde{y}_i) \cdot \frac{x}{\alpha_i \frac{1}{n}\sum_{k=1}^{n}\exp(\tilde{y}_k)} \end{aligned}$$

We note that in our case we do not use a weighted formulation, since the ERA5 LR data is created by average pooling without weighting and the WRF data covers a small area, so there the lat-lon cells have about the same area.

3.3.3 Relaxing constraints and global constraining

The constraint layers can be relaxed by increasing the constraint window size; this can then impose soft constraints. In the extreme case, this would reduce the number of constraints to one and gives the possibility of adding global constraint. The constraints would be the same as in Eq. (1), but with n being the number of total pixels.

3.3.4 Application in emulation

Our constraining methodology is not limited to downscaling and can enforce mass conservation e.g. in emulation tasks. An example could be aerosol microphysics emulation (Harder et al.,

2022), where different aerosol masses need to be conserved within each time step. The predicted aerosol masses among different size bins $y_i, i \in I_{\text{dust}}$ for a specific aerosol type, eg. dust, have to add up to the sum of the input aerosol masses $X_i, i \in I_{dust}$ of the same species:

$$\sum_{i \in I_{\text{dust}}} y_i = \sum_{i \in I_{\text{dust}}} X_i$$

This conservation of mass can be enforced with the AddCL, MultCl, or SmCl:

$$\begin{aligned} y_i^{\text{AddCL}} &= \tilde{y}_i + \sum_{k \in I_{\text{dust}}} X_k - \sum_{k \in I_{\text{dust}}} \tilde{y}_k \\ y_i^{\text{MultCL}} &= \tilde{y}_i \cdot \frac{\sum_{k \in I_{\text{dust}}} X_k}{\sum_{k \in I_{\text{dust}}} \tilde{y}_k} \\ y_i^{\text{SmCL}} &= \exp(\tilde{y}_i) \cdot \frac{\sum_{k \in I_{\text{dust}}} X_k}{\sum_{k \in I_{\text{dust}}} \exp(\tilde{y}_k)} \end{aligned}$$

Here, SmCL again would additionally guarantee positive masses.

4. Data

To test and evaluate our proposed method, we create a variety of data sets as well as use existing and established ones. We generate multiple data sets based on the ERA5 data using average pooling to create the LR inputs, which has been the standard methodology in climate downscaling studies (see e.g. Serifi et al. (2021); Leinonen et al. (2021)). We also use data sets based on the outputs of models such as the Weather and Research Forecasting (WRF) Model and the Norwegian Earth System Model (NorESM) that contain real low-resolution simulation data matched to high-resolution data. Finally, we test our methods on non-climate data sets: lunar satellite imagery and natural images. An overview of all the different data sets used can be found in Table 1.

4.1 ERA5 data set

The ERA5 data set (Hersbach et al., 2020) is a so-called *reanalysis* product from the ECMWF that combines model data with worldwide observations. The optimal physical model state that best fits the observations is found through the process of data assimilation. ERA5 is available as global, hourly data with a $0.25° \times 0.25°$ resolution, which is roughly 25 km per pixel in the mid-latitudes. It covers all years starting from 1950.

Total water content data set For this work, the quantity we focus on is the total column water (tcw) that is given in kg/m^2 and describes the vertical integral of the total amount of atmospheric water content, including water vapour, cloud water, and cloud ice but not precipitation.

Spatial SR data To obtain our high-resolution data points we extract a random 128×128 pixel image from each available time step (each time step is 721×1440 and there are roughly 60,000 time steps available). We randomly sample 40,000 data points for training and 10,000 for each validation and testing. The low-resolution counterparts are created by taking the

Table 1: The different data sets we use to test our constraint layers. The names are given to identify the data sets throughout the paper. Most data sets are based on ERA5 atmospheric water content data and LR is generated synthetically, we include different upsampling factors, an ood case, and temporal data sets. Additional data sets include the moist static energy (MEn) data set as well as WRF and NorESM model data. Lunar and natural images give non-climate application data sets. The results for data sets in bold can be found in the main paper the rest is given in the appendix for improved focus and clarity.

Name	Source	Type	Dim. LR/HR	Size train/val/test
TCW2	ERA5	water cont.	(1,64,64)/(1,128,128)	40k/10k/10k
TCW4	ERA5	water cont.	(1,32,32)/(1,128,128)	40k/10k/10k
TCW8	ERA5	water cont.	(1,16,16)/(1,128,128)	40k/10k/10k
TCW16	ERA5	water cont.	(1,8,8)/(1,128,128)	40k/10k/10k
TCW OOD	ERA5	water cont.	(1,32,32)/(1,128,128)	40k/10k/10k
TCW T1	ERA5	water cont.	(3,32,32)/(3,128,128)	40k/10k/10k
TCW T2	ERA5	water cont.	(2,32,32)/(3,128,128)	40k/10k/10k
MEn	ERA5	water vapor liq. water temp.	(3,32,32)/(3,128,128)	40k/10k/10k
WRF	WRF	temp.	(1,45,45)/(1,135,135)	20k/4k/4k
NorESM	NorESM	temp.	(1,32,32)/(1,64,64)	24k/12k/12k
Lunar	satell.	photons	(1,32,32)/(1,128,128)	132k/16k/16k
Nat	Nat. images	RGB	(3,128,128)/(3,512,512)	var.

mean over $N \times N$ patches, where N is our upsampling factor. A sample pair is shown in Figure 3 a). This operation is physically sound, considering that conservation of water content means that the water content (density per squared meter) described in an LR pixel should be equal to the average of the corresponding HR pixels. We can also observe in LR-modeled data such as WRF data (see below) that the modeled quantities in a low-resolution run are approximately the mean of a high-resolution run, which further justifies our coarsening strategy.

Spatio-Temporal data sets Including the temporal evolution of our data, we create two additional data sets. For the first data set, one sample consists of 3 successive time steps, the same time steps for both input and target, but at different resolutions. This is done to perform spatial SR for multiple time steps simultaneously, see Figure 3 b). We select three random 128×128 pixel areas per global image, resulting in the same number of examples as the procedure described above. We split the data randomly as before, and each time step is downsampled by taking the spatial mean. We then create a second data set, that is built for the learning task of increasing both spatial and temporal dimensions. We again crop three images out of a series of three successive time steps to obtain our high-resolution target. To create the low-resolution input, we decrease both temporal and spatial dimensions. To

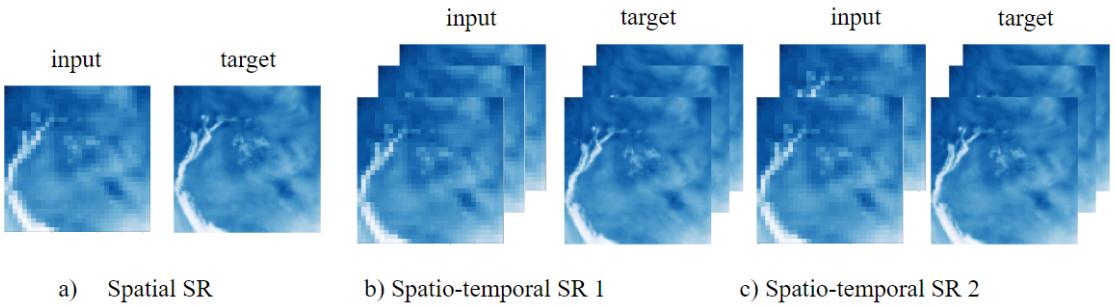

Figure 3: Samples of the three different data set types used in this work. a) A data pair we use for our standard spatial super-resolution task. The input is an LR image and the target is the HR version of that. b) A data pair for performing SR for multiple time steps simultaneously. The input is a time series of LR images and the output is the same time series in HR. c) A data pair where SR is performed both temporally and spatially, with two LR time steps as input and 3 HR time steps as a target.

decrease the temporal resolution, we remove the intermediate (the second) time step in each sample, i.e. perform sub-sampling. To decrease the spatial resolution we apply the same operation as before, i.e. compute the mean spatially. These results result in two LR inputs, see Figure 3 c). Temporally coarse-graining by subsampling not by averaging is done to avoid leakage of future information into previous time steps

OOD data set For the data sets described above, the train-val-test split is done randomly. To understand how our constraining influences out-of-distribution generalization, we create a data set with a split in time. Here, we expect patterns to appear in the later time steps that are out-of-distribution of what was previously observed. We train on older data and then test on more recent years: for training, we use the years 1950-2000, for validation 2001-2010, and for final testing 2011-2020.

Energy data set Also originating from the ERA5 data, we create a second data set including different physical variables coming with different constraints as well. This data set is constructed to preserve moist static energy and water masses while predicting water vapor, liquid water content, and air temperature. The variables are taken from the pressure level at 850hPa.

4.2 WRF data

In Watson et al. (2020), a data set using the Advanced Research version of the WRF Model is introduced. It comprises hourly operational weather forecast data for Lake George in New York, USA from 2017-01-01 to 2020-03-20. More details about the model and its configuration can be found in Watson et al. (2020). The variable we consider for this work is the temperature at 2m above the ground. Unlike the previous data sets, this one does not involve synthetic downsampling but includes two forecasts run at different resolutions with

different physics-based parameterizations: one at 9 km horizontal resolution and one at 3 km. Our goal is to predict the 3 km resolution temperature field given the 9 km one and builds on work by Auger et al. (2021), which used the same data set.

4.3 Constraints in our data sets

In predicting distinct physical quantities, there are different constraints we need to consider. Most of our data sets include the downscaling constraints given by (1), which are satisfied by the LR-HR pairs either approximately (for simulations that are run at LR and HR with quantities respecting physical conservation laws) or exactly (in the case of average pooling for creating the LR version). We detail the constraints in the following subsections.

Water content conservation For predicting the total column-integrated water content, we are given the low-resolution water content $Q^{(LR)}$ and must obtain the super-resolved version $Q^{(SR)}$. The downscaling constraint or mass conservation constraint (1) for each LR pixel $q^{(LR)}$ and the corresponding super-pixel $(q_i^{(SR)})_{i=1,\dots,n}$ is then given by

$$\frac{1}{n}\sum_{i=1}^{n} q_i^{(SR)} = q^{(LR)}. \tag{10}$$

Moist static energy conservation One of our tasks includes predicting column-integrated water vapor, liquid water, and temperature while conserving both water mass and moist static energy. As described above, water mass conservation is straightforward, directly applying our constraining methodology. On the other hand, the (column-integrated) moist static energy S is approximated by:

$$S \approx ((1 - Q_v) \cdot c_{pd} + Q_L \cdot c_l) \cdot T + L_v \cdot Q_v, \tag{11}$$

where

$$L_v \approx 2.5008 \cdot 10^6 + (c_{pw} - c_L) \cdot (T - 273.16)$$

is the latent heat of vaporization in (Jkg^{-1}). The water vapor $Q_v[kg \cdot kg^{-1}]$, the liquid water $Q_L[kg \cdot kg^{-1}]$, and the temperature $T[K]$ are being predicted, whereas c_{pd}, c_{pv} and $c_L[J \cdot K^{-1} \cdot kg^{-1}]$ are heat capacity constants.

We use the following procedure to predict these quantities while conserving moist static energy:

1. Given LR $T^{LR}, Q_V^{LR}, Q_L^{LR}$
2. Calculate LR S^{LR} with (11)
3. Predict SR $S^{SR}, Q_v^{SR}, Q_L^{SR}$ while enforcing (1) using one of our constraint layers
4. Calculate SR T^{SR} using (11) and SR $S^{SR}, Q_v^{SR}, Q_L^{SR}$.

This means we predict T^{SR} not directly, but by predicting S^{SR}. We are then able to predict the temperature T while ensuring (approximate) energy conservation by applying our constraint layer to the prediction of S^{SR}.

Different simulations If the LR-HR pairs are not created by taking the local mean of the HR but by using two simulations run at different resolutions, the downscaling constraint is not automatically satisfied in the data. This is the case for our WRF and NorESM data sets (NorESM data is discussed in the appendix; here, we focus on WRF). Even though the downscaling constraint is not exactly obeyed (see Figure 4), it is approximately, and we can still apply our constraining in the same way as before. If the real low-resolution data and the downsampled high-resolution data are not significantly dissimilar, constraining can still benefit the predictive ability.

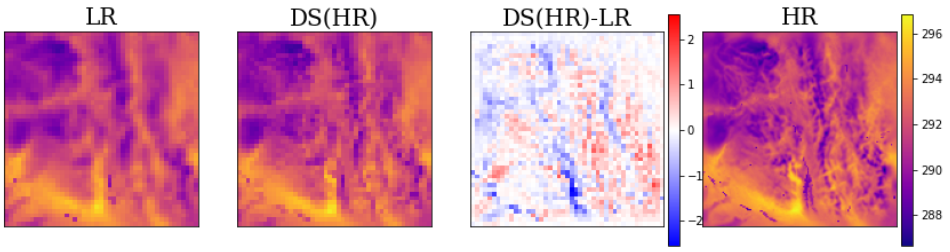

Figure 4: A LR-HR pair from the WRF temperature data. HR and LR come from different runs using the same model at different resolutions. Here we compare the real LR with the low-resolution data created by average pooling of the HR, written as DS(HR). It shows that there is not an exact match between LR and downsampled HR, which makes the success of a constraint layer more difficult. The violation of the downscaling constraint in the WRF data set is 0.684 on average.

5. Experimental setup

We conduct two sets of experiments:

1. Show the applicability of our constraining method to different neural network architectures.
2. Show the applicability of our constraining method to different data sets and different constraint types.

In most of our experiments, we use synthetic low-resolution data created by applying average pooling to the original high-res samples, as is usually done to test perfect prognosis downscaling setups. Additionally, we consider cases with pairs of real low-res and high-res simulations to show that our methods work in the intended final application.

5.1 Architectures

We test our constraint methods throughout a variety of standard deep learning SR architectures including an SR CNN, conditional GAN, a combination of an RNN and CNN for spatio-temporal SR, and a new architecture combining optical flow with CNNs/RNNs to

increase the resolution of the temporal dimension. The original, unconstrained versions of these architectures then also serves as a comparison for our constraining methodologies.

SR-CNNs Our SR CNN network, similar to Lim et al. (2017), consists of convolutional layers using 3×3 kernels and ReLU activations. The upsampling is performed by a transpose convolution followed by residual blocks (convolution, ReLU, convolution, adding the input, ReLU). The architecture for $2\times$ downscaling is shown in Figure 2.

SR-GAN A conditional GAN architecture (Mirza and Osindero, 2014) is a common choice for super-resolution (Ledig et al., 2016). Our version uses the above-introduced CNN architecture as the generator network. The discriminator is used from (Ledig et al., 2016), it consists of convolutional layers with a stride of 2 to decrease the dimensionality in each step, with ReLU activation. It is trained as a classifier to distinguish SR images from real HR images using a binary cross-entropy loss. The generator takes as input both Gaussian noise as well as the LR data and then generates an SR output. It is trained with a combination of an MSE loss, helping reconstruction, and the adversarial loss given by the discriminator, like a standard SR GAN, e.g. Ledig et al. (2017).

SR-ConvGRU We apply an SR architecture based on the GAN presented by Leinonen et al. (2021), which uses ConvGRU layers to address the spatio-temporal nature of super-resolving a time series of climate data. Here, we use the generator on its own, both during inference and training time without the discriminator, providing a deterministic approach.

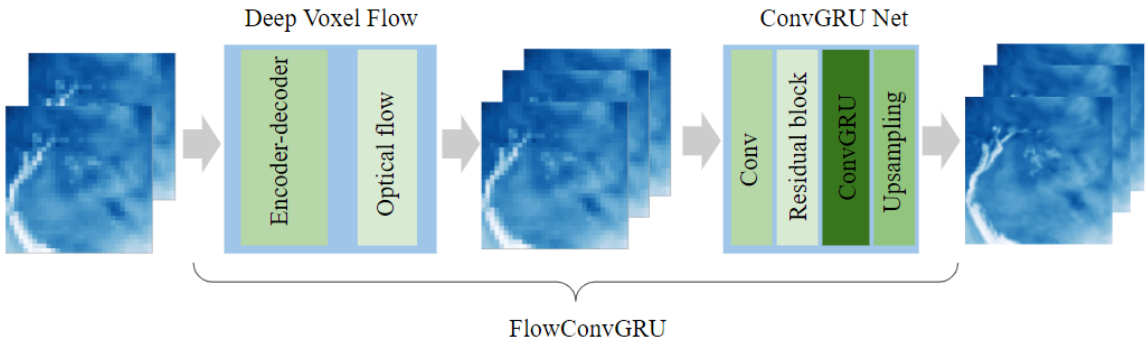

Figure 5: Our novel spatio-temporal architecture, combining Deep Voxel Flow and a ConvGRU. The inputs are two LR images at two times, the first part predicts the in-between time step using the Deep Voxel Flow model, the second part increases the spatial resolution of the three time steps using a Convolutional GRU net.

SR-FlowConvGRU To increase the temporal resolution of our data we employ the Deep Flow method (Liu et al., 2017), a deep learning architecture for video frame interpolation combining optical flow methods with neural networks. We introduce a new architecture combining the Deep Flow model and the ConvGRU network (FlowConvGRU): First, we increase the temporal resolution resulting in a higher-frequency time-series of LR images on which we then apply the ConvGRU architecture to increase the spatial resolution. The

combined neural networks are then trained end-to-end. The architecture is shown in Figure 5.

5.2 Training

Our models were trained with the Adam optimizer, a learning rate of 0.001, and a batch size of 256. We trained for 200 epochs, which took about 3—6 hours on a single NVIDIA A100 Tensor Core GPU, depending on the architecture. All models use the MSE as their criterion, the GAN additionally uses its discriminator loss term. All the data are normalized between 0 and 1 for training, except for the cases where the ScAddCL is applied. In the case of this constraint layer we scale the data between -1 and 1 as proposed in Geiss and Hardin (2023). For our time-dependent models though, ConvGRU and FlowConvGRU, we are scaling between 0 and 1, because the original scaling led to NaN-values during training.

5.3 Baselines

Pixel enlargement This baseline consists of scaling the LR input to the same size as the HR by duplicating the pixels. We include this to have reference metrics that reflect how close the LR is to the HR data. This baseline conserves mass by construction.

Bicubic upsampling As a simple non-ML baseline, we use bicubic interpolation for spatial SR and take the mean of two frames for temporal SR.

Soft constraining Soft-constraining has been successfully applied before to a variety of physics-informed deep-learning tasks. Here we use it to see how it compares to hard constraints. Soft-constraining is done by adding a regularization term to the loss function. Our MSE loss is then changed to the following:

$$\text{Loss} = (1-\alpha)\cdot \text{MSE} + \alpha \cdot \text{Constraint violation}, \tag{12}$$

where the constraint violation is the mean overall constraint violations between an input pixel x and the corresponding super-pixel $y_i, i = 1, \ldots, n$:

$$\text{Constraint violation} = \text{MSE}\left(\frac{1}{n}\sum_{i=1}^{n} y_i,\ x\right). \tag{13}$$

We conducted an experiment to investigate the impact of α values on final model performance; the results are reported in the appendix. For our main paper we choose $\alpha = 0.99$.

Unconstrained counterparts Furthermore, we always compare against an unconstrained version of the above-introduced standard SR NN architectures (SR-CNN, SR-GAN, SR-ConvGRU, SR-FlowConvGRU).

Clipping We also run the standard CNN, but with clipping applied at inference. This is a common practice to remove negative values. Results can be found in the appendix, see Table 4. This method does not guarantee mass conservation nor significantly improves performance.

6. Results and discussion

For evaluating our results, we use typical metrics for weather and climate super-resolution: root-mean-square error (RMSE), mean absolute error (MAE) and mean bias as well as typical metrics for super-resolution: peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), multi-scale SSIM (MS-SSIM), Pearson correlation and Fractional Skill Score (FSS). We show RMSE and MS-SSIM in the main paper, while the others can be found in the appendix. Most metrics are highly correlated in our case. For the GAN giving a probabilistic prediction, we also use continuous ranked probability score (CRPS). Because we are interested in the violation of conservation laws and predicting non-physical values, we also look at the average constraint violation, the number of (unwanted) negative pixels, and the average magnitude of negative values. We additionally look at the variance among the pixels within a predicted super-pixel and investigate the difference for constraining methods. The key results are aggregated in Figure 6.

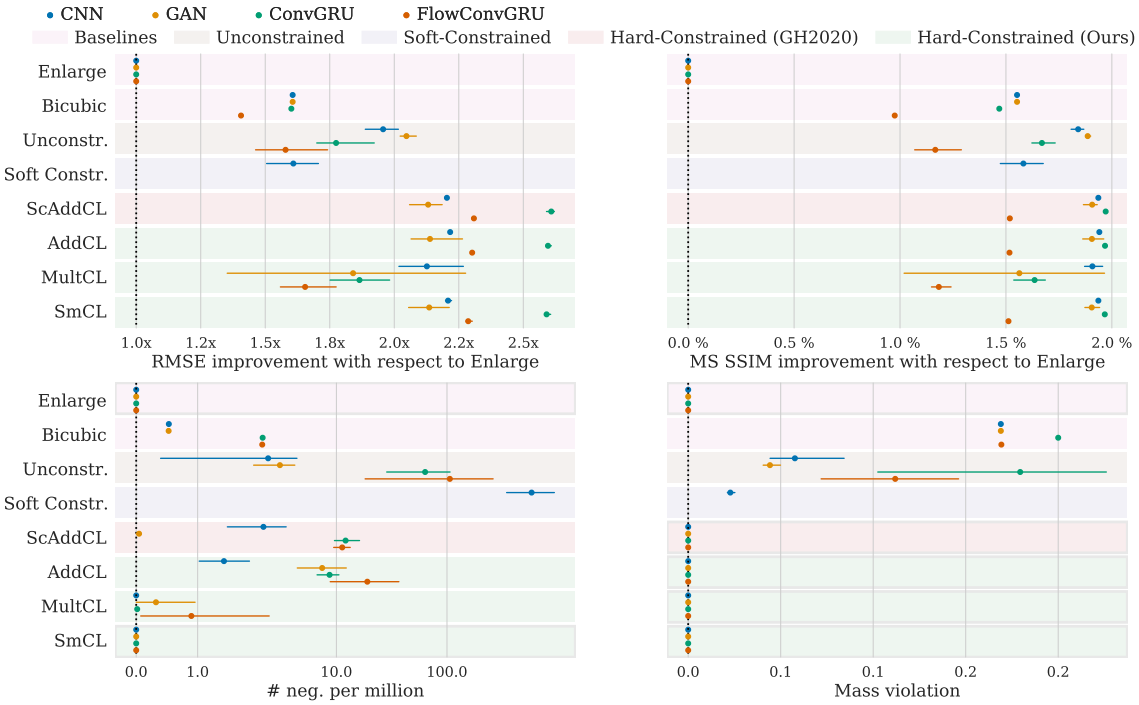

Figure 6: Metrics for different constraining methods and architectures applied to the water content data sets (TCW4, TCW T1 and TCW T2), calculated over 10,000 test samples. The mean and confidence interval from 3 runs are shown, for RMSE and MS-SSIM relative to the Enlarge baseline for number of negative pixels (per mil.) and mass conservation violation the absolute values are shown. The framed box indicates that the method achieves zero violation of the physics, no negative pixels or mass conservation up to numerical precision. Tables with more metrics can be found in the appendix

6.1 Different constraining methods

Whereas hard-constraining shows exact conservation and appears to enhance performance, the application of soft-constraining on the other hand does decrease constraint violation, but still maintains a significant magnitude of it, which can be seen in Figure 6 for example. Also, soft-constraining seems to suffer from an accuracy-constraints trade-off, where depending on the regularization factor α, either the constraint violation is reduced, or the accuracy increases, but it struggles to do both simultaneously. A table for different α is shown in the appendix. Among the hard-constraining methodologies, the multiplicative renormalization layer, MultCL, performs the weakest in terms of predictive skills (see Figure 6), which could be due to instability when inputs get close to zero. The three other methods, ScAddCL, AddCL, and SmCL, often have very similar measurements. SmCL shows the advantage of also enforcing positivity when necessary (see Figure 6). ScAddCL divides the number of violation by more than 2 compare to the AddCL and MulCL gets close to zero violation in many cases.

6.2 Different architectures

As shown in Figure 6 for all architectures (CNN, GAN, ConvGRU, FlowConvGRU), adding the constraint layers enforces the constraint and improves the evaluation metrics compared to the CNN case. Constraining the GAN leads to less of a performance boost, but AddCL and SmCL still enhance the predictions compared to the unconstrained GAN. Including the temporal dimensions, the constraining improves the prediction quality much more significantly than in the case with just a single time step (see Figure 6).

6.3 Different data sets and constraints

The success of our constraining methodology does not depend on the upsampling factor: in Table 5, we can see that the constraining methods work well and improve all metrics for upsampling factors of 2, 4, 8, and 16. When applied to our out-of-distribution data set, the improvement achieved by adding constraints is even more pronounced than for the randomly split data (see results in the appendix). The constraints can help architectures with their generalization ability.

Not only mass can be conserved, but other quantities such as moist static energy. We show that moving on to different quantities of the ERA5 data set, temperature, water vapor, and liquid water. Looking at Table 10 (see appendix), one can observe similar results for liquid water Q_L and water vapor Q_v as for the total water content: ScAddCL, AddCL, and SmCL significantly improve results in all measures over the unconstrained CNN, while enforcing energy and mass conservation. For temperature, on the other hand, MultCL performs the strongest, followed by SmCL, whereas AddCL and ScAddCL achieve smaller improvements in the scores.

Our WRF temperature data set includes low-resolution data points drawn from a separate simulation, rather than downsampling, and therefore it results in much harder tasks. Table 2 shows that the scores are improved slightly with our constraint layer, this might be counterintuitive given there is a violation in the training data, but this violation is relatively small, it appears like random noise, so no bias is introduced. This way the constraints again

lead to a simpler learning problem and are able to improve performance. The fact, that the constraints are slightly violated in the original data set could motivate soft-constraining, but nevertheless, we can observe that soft-constraining harms the predictive performance, while hard-constraining is surprisingly beneficial. The constraint violation in the original data has an RMSE of 0.6838 on average.

Table 2: We show four metrics for different constraining methods applied to the SR CNN applied on the WRF temperature data, calculated over 10,000 test samples. We choose the most common (RMSE, MAE, SSIM) and relevant (constr. viol) for our cases. The mean is taken over 3 runs. The best scores are highlighted in bold blue.

Data	Model	Constraint	RMSE	MAE	MS-SSIM	Constr. viol.
WRF	Enlarge	none	1.015	0.648	94.51	**0.000**
WRF	CNN	none	0.952	0.618	94.92	0.181
WRF	CNN	soft	1.020	0.660	94.57	0.032
WRF	CNN	SmCL	**0.950**	**0.592**	**95.25**	**0.000**

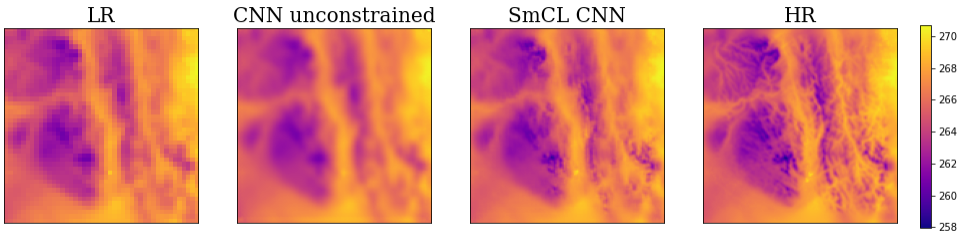

Figure 7: A random prediction for the WRF temperature test data set. We compare unconstrained and softmax-constrained predictions. It can be seen that in this case, the constraining improves the visual quality significantly including more fine-grain details.

Finally, we also show that applying our constraint methodology can improve results in other domains, even in cases where there is no physics involved. We see that both for the lunar satellite imagery and the natural images benchmark data sets, the application of our SmCL improves the traditional metrics, as shown in Tables 15 and 16.

6.4 Perceptual quality of predictions

Additionally to an enhancement quantitatively, we can see an improved visual quality for some examples, as shown in Figure 8 and 9 for the water content data. For the WRF temperature forecast data, we see a very significant improvement in the perceptual quality of the prediction. Looking at an example, such as shown in Figure 7, we can see how much

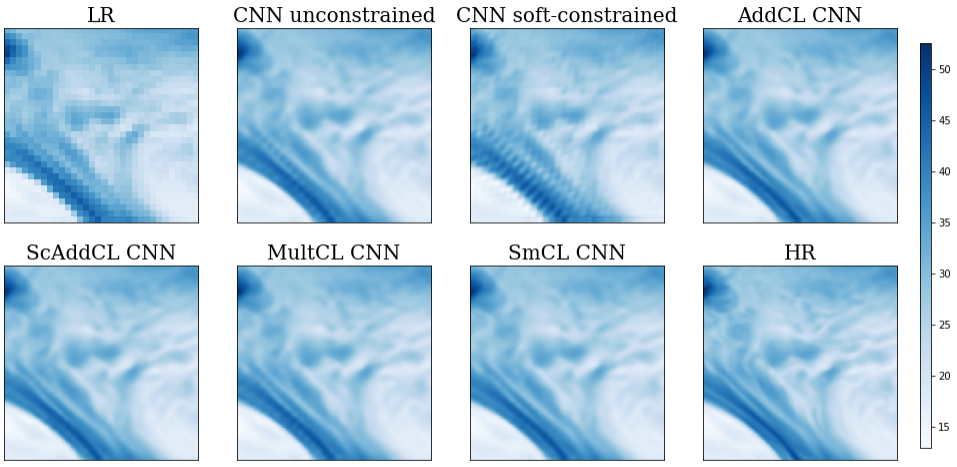

Figure 8: One example image from the test set. Shown here are the LR input, different constrained and unconstrained predictions, and the HR image as a reference. This example is from the TCW4 test data set. For the unconstrained CNN prediction, we can observe some artifacts in the lower left part, which get amplified by applying soft-constraining but decreased using hard-constraining like AddCL, ScAddCL, or SmCl.

more detail is added to the prediction when adding our constraining. For the lunar satellite imagery, Figure 19 shows that applying constraints can make the image slightly less blurry.

6.5 Development of error during training

Observing how the MSE develops during training (see Figure 10), we can see that the curve of the constrained network is generally lower than the unconstrained one. Additionally, it can be seen that constraining helps smooth both the training and validation curves.

6.6 Spatial distribution of errors

A known issue in downscaling methods is the so-called coastal effect, where errors of predictions tend to be more pronounced in coastal regions. Besides coastal region areas, mountain ridges can also be critical. In Figure 11, we show the error of the unconstrained prediction for water content and the softmax-constrained prediction. We can see that both predictions show more errors in coastal and mountainous regions. However, if we analyze the difference in errors between the unconstrained and constrained versions, we can see in Figure 12 that constraining leads to lower errors in those areas.

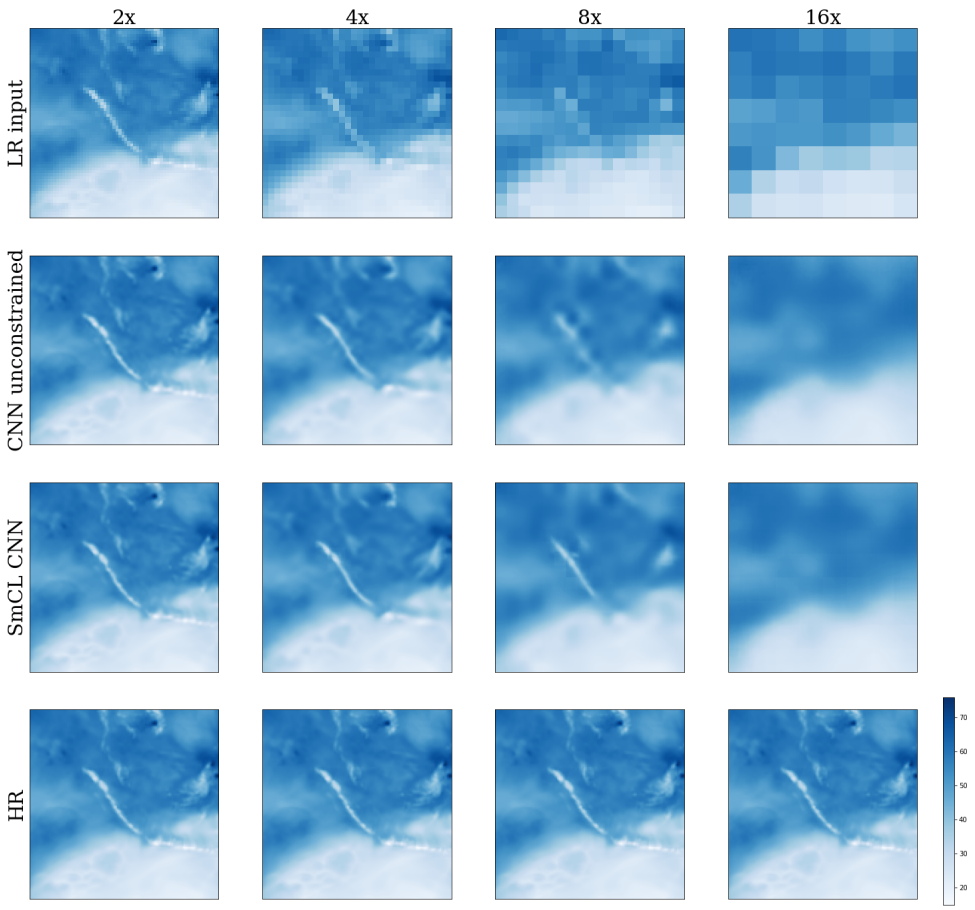

Figure 9: One example image is chosen randomly from the test set. Each model was trained for the same target resolution but with a different upsampling factor. The first row shows the LR inputs for each resolution and the last row the corresponding HR ground truth. The second and third rows show the prediction of an unconstrained CNN and with the SmCL, respectively.

6.7 Limitations

In the case of our WRF data set, we have seen that the constraining methodology can improve predictive performance even if the underlying constraints are slightly violated by the original data. In cases where low-resolution and its high-resolution counterpart are too far apart, our model is not always able to increase the predictive skill. We built a data set from two

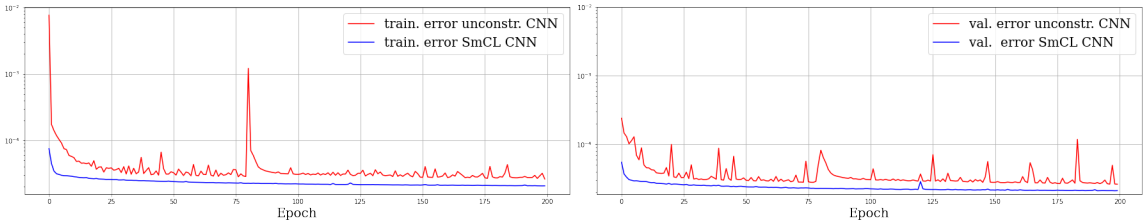

Figure 10: The development of training and validation errors with increasing iterations during training. Shown for an unconstrained CNN and CNN+SmCL applied to the water content data. We can observe how hard constraining accelerates convergence and smooths the learning curve, both measured in training and validation error.

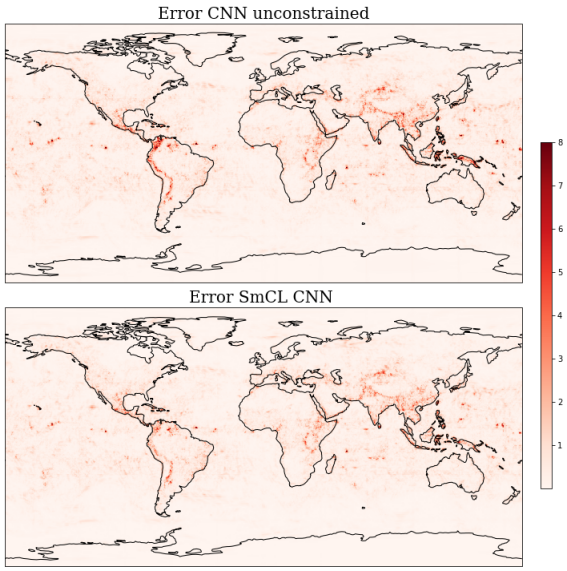

Figure 11: The errors of the global predictions for unconstrained and constrained (SmCL) CNNs, when compared to the ground truth. The CNN is applied per 32×32 patch and then put together for a global predictions at a random time step. Used here is the TCW4 data set. We can observe how the stronger errors in coastal and mountainous regions for the unconstrained predictions are dampened by soft-max constraining.

different resolutions of the Norwegian Earth System Model (NorESM) (Seland et al., 2020), and applying our constraining methods improved the visual similarity of the predictions, but decreased the predictive ability. We provide scores and plots in the appendix. In the case of other sampling strategies such as subsampling spatially, our methods are not applicable in their current form and they depend on having constraints that can be formulated with Eq. (8).

7. Conclusion and future work

This work presents a novel methodology to incorporate physics-inspired downscaling hard constraints into neural network architectures for climate super-resolution. We show that this method performs well across different deep learning architectures, upsampling factors, predicted quantities, and data sets. We demonstrate its effectiveness both on standard downscaling data sets and on data created by independent simulations. Our constrained models are not only guaranteed to satisfy consistency such as mass conservation between LR and HR, but also increase predictive performance across metrics and use cases. Compared to soft-constraining through the loss function, our methodology does not suffer from the common accuracy-constraints enforcement trade-off. Our hard-constraining performance enhancement is not only limited to climate super-resolution but also noticeable in satellite imagery of the lunar surface as well as standard benchmark data sets of natural images. Within the climate context, our constraint layer can help with common issues connected to deep learning applied to downscaling: it dampens the coastal effect, errors get lower in critical regions, out-of-distribution generalization is improved and training can be more stable. Hard-constraining can weaken performance if the enforced relationships are strongly violated in the true data (see NorESM data). If a bias exists in the LR (or other input) it can be propagated to the HR prediction by constraining on the LR.

Future work could extend the application of our constraint layer to other climate-related tasks beyond downscaling. Climate model emulation (e.g. Beucler et al. (2021) and Harder et al. (2021)) for example could strongly benefit from a reliable and performance-enhancing method to enforce physical laws. For post-processing purposes, the offline application of our method, our code is readily available. To deploy these constrained super-resolution methods online, the next step is to use Fortran-Python bridges (e.g. (Ott et al., 2020)) to include them in global climate model runs.

Acknowledgement and Disclosure of Funding

PH acknowledges the funding received by the Fraunhofer Institute for Industrial Mathematics. DR was funded in part by the Canada CIFAR AI Chairs Program. The authors also are grateful for support from the NSERC Discovery Grants program, material support from NVIDIA in the form of computational resources, and technical support from the Mila IDT team in maintaining the Mila Compute Cluster.

Appendix A: Tuning soft-constraining

Here we investigate the influence of the factor α on the soft-constraining method in more detail. Table 3 shows how the increase of α improves the mass conservation but only up to a value between 0.014 and 0.017. At the same time, it shows that the predictive skill decreases with the increase of α significantly.

Table 3: Metrics calculated over 10,000 validation samples. The best scores are highlighted in bold blue, second best in bold black.

Data	Alpha	RMSE	MAE	MS-SSIM	Mass viol.	#Neg per mil.
TCW4	0.0001	**0.241**	**0.102**	99.95	0.021	1.21
TCW4	0.001	**0.237**	**0.100**	**99.96**	0.022	**0.12**
TCW4	0.01	0.247	0.103	**99.96**	0.022	1.39
TCW4	0.1	0.252	0.104	99.95	0.023	**0.41**
TCW4	0.9	0.268	0.110	99.95	0.020	16.83
TCW4	0.99	0.297	0.133	99.94	**0.014**	31.01
TCW4	0.999	0.477	0.261	99.84	**0.016**	600.96
TCW4	0.9999	0.706	0.433	99.71	0.017	3867.90
TCW4	1	2.618	1.814	94.22	0.017	960.42

Appendix B: Clipping for nonnegativity

As natural RGB images have a well-defined range, it is common in CNN and GAN implementations to clip the pixels at inference time to the desired range, removing negative values, for example. Here, in Table 4 we show that doing that gives a very small increase in performance, but still performs significantly worse than SmCL, which achieves also zero negative values. We want to point out that a combination of a constraint layer such as MultCL and clipping would lead to the clipping layer to destroy the enforced consistency given by the contraint layer if applied afterwards.

Table 4: Metrics for different constraining methods applied to the SR CNN + clipping applied on the water content data set, calculated over 10,000 test samples. The mean is taken over 3 runs. The best scores are highlighted in bold blue.

Data	Model	Constraint	RMSE	MAE	MS-SSIM	Mass viol. per mil.	#Neg
TCW4	CNN	none	0.661	0.327	99.39	0.059	2.41
TCW4	CNN	clip	0.657	0.326	99.440	0.058	**0**
TCW4	CNN	SmCL	**0.582**	**0.291**	**99.49**	**0.000**	**0**

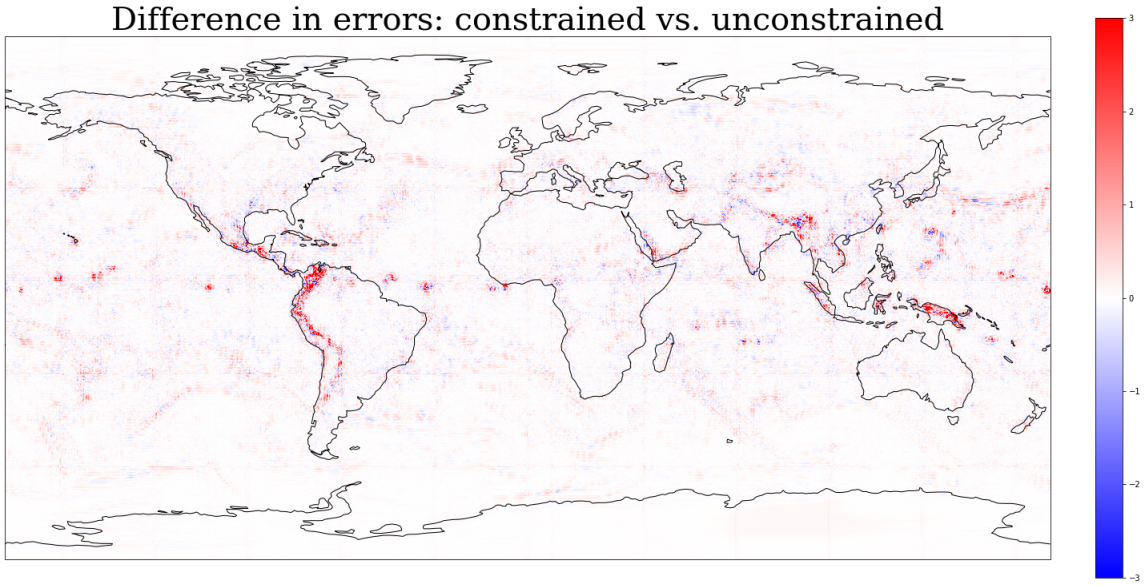

Figure 12: The difference in the errors of constrained and unconstrained predictions from Figure 11. Positive values (red) mean a higher error in the unconstrained version. We trim values at 3, so everything that has a difference greater than 3 is shown as full red for better visibility.

Appendix C: Score tables

We show the tables with the mean scores that are displayed as Figures in the main paper and additionally include the MAE.

One observation from TFigure 13 is that the RMSE improvement is better for lower upsampling factors but the other way around for MS SSIM. A potential explanation: For higher upsampling factors it gets increasingly difficult to achieve good visual (read high SSIM) quality, whereas the RMSE is still relatively easy to minimize. Here, adding the constraint layers have more leverage to improve.

Appendix D: Additional scores

We look at additional scores for our water content data set. We investigate the mean bias (mean over the difference for each pixel value of prediction and truth), the peak signal-to-noise ratio (PSNR), the structural similarity index measure, the Pearson correlation (Corr), and the negative mean (the average magnitude of predicted negative values, the average is calculated over all predicted values, including positive, that are set to zero to calculate the negative mean). These metrics show a similar trend then the metrics shown in the

Table 5: Metrics for different constraining methods applied to an SR CNN, calculated over 10,000 test samples of the water content data. The mean is taken over 3 runs. The best scores are highlighted in bold blue, second best in bold.

Data	Fact.	Model	Constraint	RMSE	MAE	MS-SSIM per mil.	Mass viol.	#Neg
TCW2	2x	Enlarge	none	0.422	0.361	99.61	**0.000**	**0**
TCW2	2x	Bicubic	none	0.322	0.137	99.90	0.066	0.25
TCW2	2x	CNN	none	0.251	0.105	99.95	0.026	1.40
TCW2	2x	CNN	soft	0.301	0.137	99.23	0.016	104.65
TCW2	2x	CNN	AddCL	0.216	**0.092**	**99.96**	**0.000**	1.31
TCW2	2x	CNN	ScAddCL	**0.199**	**0.0876**	**99.96**	**0.000**	0.02
TCW2	2x	CNN	MultCL	0.223	0.094	**99.96**	**0.000**	**0**
TCW2	2x	CNN	SmCL	**0.215**	0.094	**99.96**	**0.000**	**0**
TCW4	4x	Enlarge	none	1.286	0.717	97.60	**0.000**	**0**
TCW4	4x	Bicubic	none	0.800	0.401	99.12	0.169	0.53
TCW4	4x	CNN	none	0.657	0.326	99.40	0.058	2.41
TCW4	4x	CNN	soft	0.801	0.410	99.15	0.023	581.54
TCW4	4x	CNN	AddCL	**0.580**	**0.290**	**99.50**	**0.000**	1.42
TCW4	4x	CNN	ScAddCL	**0.575**	**0.289**	**99.50**	**0.000**	0.07
TCW4	4x	CNN	MultCL	0.606	0.300	99.47	**0.000**	**0**
TCW4	4x	CNN	SmCL	0.582	0.291	99.49	**0.000**	**0**
TCW8	8x	Enlarge	none	2.181	1.294	92.39	**0.000**	**0**
TCW8	8x	Bicubic	none	1.557	0.900	96.49	0.318	6.56
TCW8	8x	CNN	none	1.358	0.782	97.15	0.109	15.48
TCW8	8x	CNN	soft	1.640	0.965	96.06	0.029	103,702
TCW8	8x	CNN	AddCL	**1.267**	**0.733**	**97.41**	**0.000**	632.32
TCW8	8x	CNN	ScAddCL	**1.264**	0.734	**97.41**	**0.000**	0.15
TCW8	8x	CNN	MultCL	1.331	**0.733**	97.22	**0.000**	0.10
TCW8	8x	CNN	SmCL	1.268	0.734	97.40	**0.000**	**0**
TCW16	16x	Enlarge	none	3.425	2.159	85.55	**0.000**	**0**
TCW16	16x	Bicubic	none	2.723	1.730	91.72	0.510	53.67
TCW16	16x	CNN	none	2.450	1.545	92.68	0.203	4.15
TCW16	16x	CNN	soft	2.794	1.776	90.74	0.036	2250.77
TCW16	16x	CNN	AddCL	**2.364**	**1.491**	**92.96**	**0.000**	457.34
TCW16	16x	CNN	ScAddCL	**2.368**	1.495	92.94	**0.000**	2.12
TCW16	16x	CNN	MultCL	2.409	1.518	92.77	**0.000**	0.17
TCW16	16x	CNN	SmCL	**2.368**	**1.492**	**92.95**	**0.000**	**0**

main paper: all of them are improved by adding constraints in our architecture. Without or with soft constraining there are small biases appearing in the predictions, but hard constraining removes those biases. PSNR is a function of the MSE and therefore shows the same trend as it. SSIM and correlation give very similar results, with ScAddCL, AddCL,

Table 6: Metrics for different constraining methods applied to an SR GAN, calculated over 10,000 test samples of the 4x upsampling water content data. The mean is taken over 3 runs. The best scores are highlighted in bold blue, and the second best in bold.

DATA	MODEL	CONSTRAINT	RMSE	MAE	CRPS	MS-SSIM PER MIL.	MASS VIOL.	#NEG
TCW4	GAN	NONE	0.628	0.313	0.1522	99.44	0.0453	3.46
TCW4	GAN	ADDCL	**0.602**	**0.306**	**0.1519**	**99.46**	**0.000**	7.38
TCW4	GAN	SCADDCL	0.604	**0.305**	**0.1508**	**99.46**	**0.000**	0.05
TCW4	GAN	MULTCL	0.732	0.406	0.1978	99.13	**0.000**	**0**
TCW4	GAN	SMCL	**0.603**	0.310	0.1520	**99.46**	**0.000**	**0**

Table 7: Metrics for different constraining methods applied to an SR ConvGRU, calculated over 10,000 test samples of the water content data. The best scores are highlighted in bold blue, second best in bold.

DATA	MODEL	CONSTRAINT	RMSE	MAE	MS-SSIM	MASS VIOL. PER MIL.	#NEG
TCW T1	ENLARGE	NONE	1.292	0.718	97.72	**0.000**	**0**
TCW T1	BICUBIC	NONE	0.807	0.402	99.16	0.169	2.16
TCW T1	CONVGRU	NONE	0.672	0.340	99.42	0.102	55.45
TCW T1	CONVGRU	ADDCL	**0.499**	**0.260**	**99.64**	**0.000**	1358.49
TCW T1	CONVGRU	SCADDCL	**0.499**	**0.260**	**99.64**	**0.000**	10.58
TCW T1	CONVGRU	MULTCL	0.903	0.472	98.98	**0.000**	0.25
TCW T1	CONVGRU	SMCL	0.500	**0.260**	**99.64**	**0.000**	**0**

and SmCL showing the best scores. Overall we can see that soft-constraining leads to the most significantly negative predictions, which would cause issues in the context of climate models and predictions.

Appendix E: Additional Visualizations

Here we present some visualizations, a prediction by the GAN (Figure 14), the FlowConvGRU (Figure 15), unconstrained and constrained example prediction from BSD100 and Urban100 (Figure 20), and a global prediction for water content (Figure 16).

Table 8: Metrics for different constraining methods applied to our FlowConvGRU, calculated over 10,000 test samples of the water content data set. The best scores are highlighted in bold blue, second best in bold.

Data	Model	Constraint	RMSE	MAE	MS-SSIM	Mass viol. per mil.	#Neg
TCW T2	Interpolation	none	0.834	0.428	99.10	0.169	2.14
TCW T2	FlowConvGRU	none	0.673	0.352	99.40	0.072	18.27
TCW T2	FlowConvGRU	AddCL	**0.509**	**0.275**	**99.63**	**0.000**	37.10
TCW T2	FlowConvGRU	ScAddCL	**0.509**	**0.274**	**99.63**	**0.000**	13.40
TCW T2	FlowConvGRU	MultCL	0.719	0.383	99.27	**0.000**	**0**
TCW T2	FlowConvGRU	SmCL	0.514	0.276	99.62	**0.000**	**0**

Table 9: Metrics for different constraining methods applied to the SR CNN applied on the OOD water content data set, calculated over 10,000 test samples. The mean is taken over 3 runs. The best scores are highlighted in bold blue.

Data	Model	Constraint	RMSE	MAE	MS-SSIM	Mass viol. per mil.	# Neg
TCW OOD	Enlarge	none	1.274	0.711	97.60	**0.000**	**0**
TCW OOD	Bicubic	none	0.792	0.397	98.63	0.167	0.55
TCW OOD	CNN	none	0.661	0.327	99.39	0.059	4.93
TCW OOD	CNN	AddCL	**0.575**	**0.287**	**99.50**	**0.000**	1.65
TCW OOD	CNN	ScAddCL	**0.573**	**0.288**	**99.50**	**0.000**	0.21
TCW OOD	CNN	MultCL	0.591	0.294	99.47	**0.000**	**0**
TCW OOD	CNN	SmCL	0.579	0.289	99.49	**0.000**	**0**

Appendix F: NorESM data

Our NorESM data set is based on the second version of the Norwegian Earth System Model (NorESM2), which is a coupled Earth System Model developed by the NorESM Climate modeling Consortium (NCC), based on the Community Earth System Model, CESM2. We build our data set on two different runs: NorESM-MM which has a 1-degree resolution for model components and NorESM2-LM which has a 2-degree resolution for atmosphere and land components. We use the temperature at the surface (tas) and a time period from 2015 to 2100. The scenarios ssp126 and ssp585 are used for training ssp370 for validation and ssp245 for testing. By cropping into 64×64 and 32×32 pixels, each scenario contains 12k data points. The results for the NorESM data are shown in Table 14: the best scores are in all cases achieved by the unconstrained CNN. This is probably due to the stronger violation

Table 10: Metrics for different constraining methods applied to the SR CNN, calculated over the test set for water vapor, liquid water, and temperature. The mean is taken over 3 runs. For Q_L, RMSE, MAE, and Constr. violation are scaled by a factor of 10^3 for readability. The best scores are highlighted in bold blue, second best in bold.

DATA	VAR.	MODEL	CONSTRAINT	RMSE	MAE	MS-SSIM	CONSTR. VIOL.
MEN	Q_v	ENLARGE	NONE	0.474	0.262	94.74	**0.000**
MEN	Q_v	BICUBIC	NONE	0.326	0.182	97.12	0.07
MEN	Q_v	CNN	NONE	0.260	0.141	98.14	0.02
MEN	Q_v	CNN	ADDCL	**0.250**	**0.133**	**98.28**	**0.00**
MEN	Q_v	CNN	SCADDCL	**0.250**	**0.133**	**98.28**	**0.00**
MEN	Q_v	CNN	MULTCL	**0.250**	**0.133**	**98.28**	**0.00**
MEN	Q_v	CNN	SMCL	**0.248**	**0.132**	**98.30**	**0.00**
MEN	Q_L	ENLARGE	NONE	0.0217	0.00862	98.34	**0.00000**
MEN	Q_L	BICUBIC	NONE	0.0186	0.00765	98.96	0.00236
MEN	Q_L	CNN	NONE	0.0157	0.00617	99.15	0.00067
MEN	Q_L	CNN	ADDCL	**0.0155**	**0.00588**	**99.18**	**0.00000**
MEN	Q_L	CNN	SCADDCL	**0.0155**	**0.00588**	**99.17**	**0.00000**
MEN	Q_L	CNN	MULTCL	0.0166	0.00647	99.06	**0.00000**
MEN	Q_L	CNN	SMCL	**0.0155**	0.00585	**99.17**	**0.00000**
MEN	T	ENLARGE	NONE	0.470	0.288	99.03	**0.0**
MEN	T	BICUBIC	NONE	0.281	0.156	99.67	159.1
MEN	T	CNN	NONE	0.459	0.287	99.03	139.7
MEN	T	CNN	ADDCL	0.276	0.160	99.67	**0.0**
MEN	T	CNN	SCADDCL	0.280	0.163	99.67	**0.0**
MEN	T	CNN	MULTCL	**0.270**	**0.155**	**99.69**	**0.0**
MEN	T	CNN	SMCL	**0.272**	**0.155**	**99.68**	**0.0**

of the downscaling constraints between low-resolution and high-resolution samples. We can see a significant difference between the real LR and the HR downsampled, as shown in Figure 18. The violation of the constraints here is 2.48 (RMSE), which is much higher than for the WRF case (0.68). The visual quality of the prediction, on the other hand, seems to be improved by constraining, an example is shown in Figure 17. One potential approach for improvements here could be lat-lon weighted constraining.

Appendix G: Non-climate data

Lunar data

Recent work (Delgano-Centeno et al., 2021) on super-resolution for lunar satellite imagery has shown how deep learning can be used to enhance the captured data to help future missions to the moon. To increase the resolution of images from regions like the south pole, where there is no high-resolution data available, a machine learning-ready data set has been created.

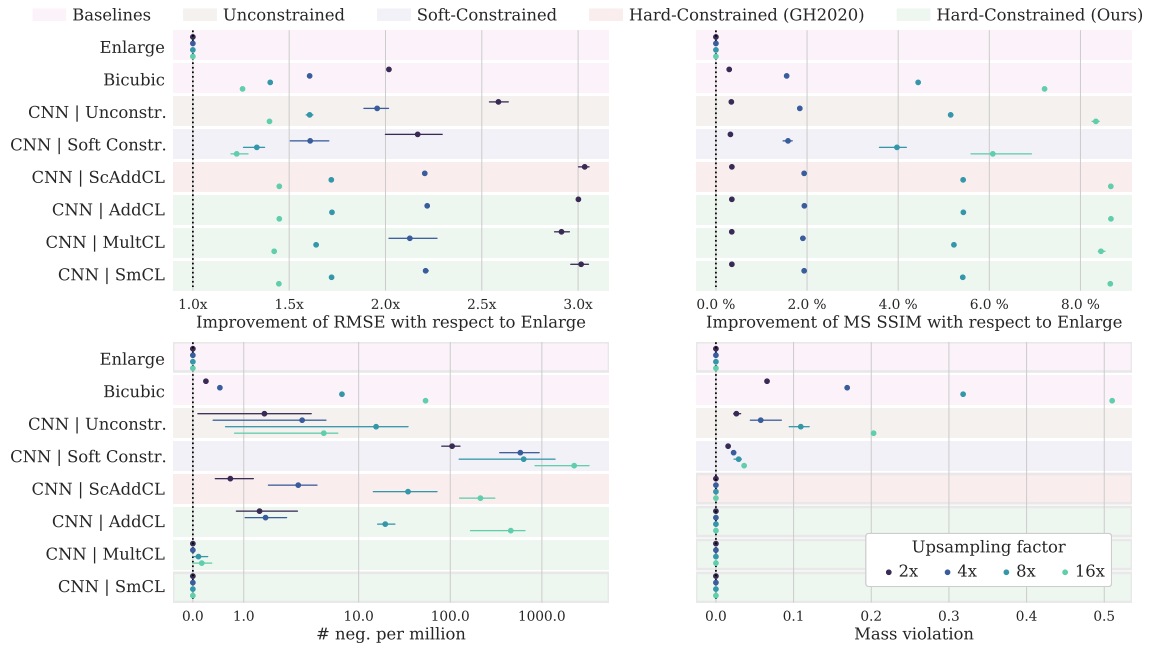

Figure 13: Metrics for different constraining methods applied to an SR CNN, calculated over 10,000 test samples of the water content data. The mean and confidence interval from 3 runs is shown relative to the Enlarge baseline. The framed box indicates a method that achieves zero violation of the physics, no negative pixels or mass conservation up to numerical precision. A table with more metrics can be found in the appendix

It consists of 220,000 images cropped out of the Narrow-Angle Camera (NAC) imagery from NASA's Lunar Reconnaissance Orbiter (LRO); for more details see Delgano-Centeno et al. (2021). Here we use a 4x upsampling version of the data set to verify if our constraining methodologies can increase the performance of super-resolution outside of climate science. The average sampling is justified in this case, because the real LR images would be created with summing photon counts in low-light regions.

Natural images

The standard benchmark data sets for super-resolution deep learning architectures applied to natural images include the OutdoorScenceTRaining (OST), DIV2K, and Flickr2k data sets for training and Set5, Set14, Urban100, and BSD100 for testing, as for example in Wang et al. (2018b). Here, we use a version resized to 512×512 pixels for HR and apply average pooling to downsample them. Our constraints depend on the downsample technique used and can not directly be applied to other downsample techniques such as sub-sampling or bicubic interpolation.

Table 11: More metrics for different constraining methods applied to an SR CNN, calculated over 10,000 test samples. The best scores are highlighted in bold blue, second best in bold.

Data	Fact.	Model	Constraint	Mean bias	PSNR	SSIM	Corr	Neg mean
TCW2	2x	Enlarge	none	**0.000**	45.36	98.65	99.75	**0.000**
TCW2	2x	Bicubic	none	**0.000**	51.46	99.71	99.95	**0.000**
TCW2	2x	CNN	none	−0.003	53.62	99.82	99.97	0.002
TCW2	2x	CNN	soft	−0.002	52.07	99.74	99.94	0.192
TCW2	2x	CNN	AddCL	**0.000**	54.91	**99.85**	**99.98**	0.002
TCW2	2x	CNN	ScAddCL	**0.000**	**55.66**	**99.87**	**99.98**	**0.000**
TCW2	2x	CNN	MultCL	**0.000**	54.65	99.84	99.97	**0.000**
TCW2	2x	CNN	SmCL	**0.000**	**54.95**	**99.85**	**99.98**	**0.000**
TCW4	4x	Enlarge	none	**0.000**	39.43	94.91	98.98	**0.000**
TCW4	4x	Bicubic	none	**0.000**	43.55	98.29	99.63	0.000
TCW4	4x	CNN	none	−0.015	45.26	98.70	99.74	0.001
TCW4	4x	CNN	soft	−0.001	43.55	98.15	99.59	0.546
TCW4	4x	CNN	AddCL	**0.000**	**46.35**	**98.89**	**99.80**	0.001
TCW4	4x	CNN	ScAddCL	**0.000**	**46.42**	**98.90**	**99.79**	**0.000**
TCW4	4x	CNN	MultCL	**0.000**	45.98	98.83	99.78	**0.000**
TCW4	4x	CNN	SmCL	**0.000**	46.31	98.88	99.79	**0.000**
TCW8	8x	Enlarge	none	**0.000**	34.84	89.08	96.95	**0.000**
TCW8	8x	Bicubic	none	+0.0001	37.77	95.40	98.50	0.006
TCW8	8x	CNN	none	−0.0148	38.96	95.93	98.82	0.012
TCW8	8x	CNN	soft	−0.0071	37.32	94.37	98.22	0.656
TCW8	8x	CNN	AddCL	**0.000**	**39.56**	**96.23**	**98.96**	0.011
TCW8	8x	CNN	ScAddCL	**0.000**	**39.58**	**96.24**	**98.97**	**0.000**
TCW8	8x	CNN	MultCL	**0.000**	39.13	95.99	98.87	**0.000**
TCW8	8x	CNN	SmCL	**0.000**	39.55	96.21	**98.96**	**0.000**
TCW16	16x	Enlarge	none	**0.000**	30.92	85.20	92.19	**0.000**
TCW16	16x	Bicubic	none	+0.0090	32.91	91.99	95.15	0.063
TCW16	16x	CNN	none	−0.0091	33.83	92.48	95.94	0.006
TCW16	16x	CNN	soft	+0.0115	32.70	90.45	94.63	4.233
TCW16	16x	CNN	AddCL	**0.000**	**34.14**	**92.67**	**96.20**	0.581
TCW16	16x	CNN	ScAddCL	**0.000**	**34.13**	**92.67**	96.18	0.007
TCW16	16x	CNN	MultCL	**0.000**	33.98	92.54	96.07	**0.000**
TCW16	16x	CNN	SmCL	**0.000**	**34.13**	**92.68**	**96.19**	**0.000**

Table 12: Fractional Skill Score (FSS) for different constraining methods and SR CNN applied on the ERA4 water content data, calculated over 10,000 test samples. We look at window sizes 2,4 and 8 and the 95th and 99th percentiles. The best scores are highlighted in bold blue.

Data	Model	Constraint	95perc.			99perc.		
			2	4	8	2	4	8
TCW4	Enlarge	none	0.970	0.989	0.997	0.935	0.974	0.991
TCW4	Bicubic	none	0.971	0.987	0.994	0.935	0.969	0.986
TCW4	CNN	none	0.978	0.992	0.997	0.950	0.979	0.993
TCW4	CNN	soft	0.971	0.989	0.997	0.935	0.974	0.991
TCW4	CNN	ScAddCL	**0.981**	**0.993**	**0.998**	**0.956**	**0.983**	**0.994**
TCW4	CNN	AddCL	**0.981**	**0.993**	**0.998**	**0.956**	**0.983**	**0.994**
TCW4	CNN	MultCL	0.979	0.992	0.998	0.951	0.980	0.993
TCW4	CNN	SmCL	**0.981**	**0.993**	**0.998**	0.955	**0.983**	**0.994**

Table 13: The variance among super-pixels for different constraining methods and SR CNN applied on the ERA4 water content data, calculated over 10,000 test samples.

Data	Model	Constraint	Variance
TCW4	Enlarge	none	0.00
TCW4	Bicubic	none	0.85
TCW4	CNN	none	1.22
TCW4	CNN	soft	0.96
TCW4	CNN	ScAddCL	1.33
TCW4	CNN	AddCL	1.32
TCW4	CNN	MultCL	1.24
TCW4	CNN	SmCL	1.34
TCW4	HR	none	1.65

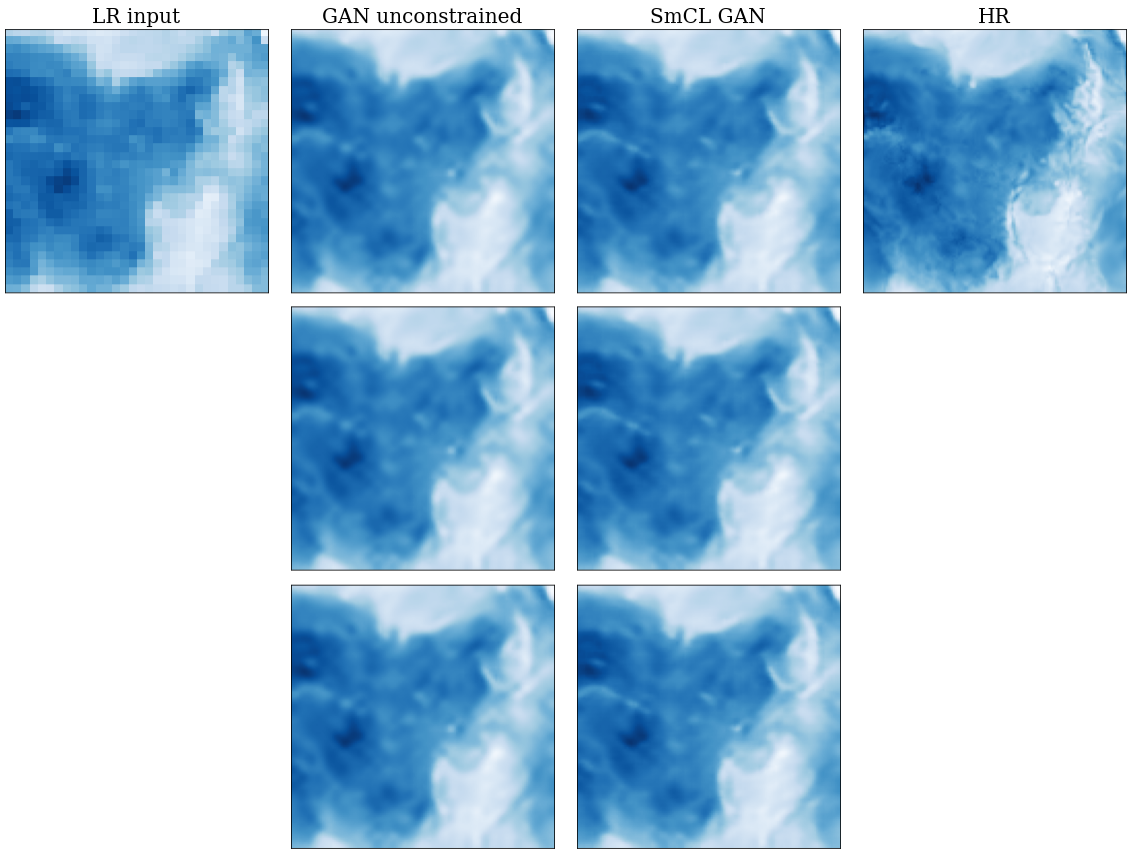

Figure 14: A random sample for the GAN predictions, showing 3 different outputs from the ensemble, constrained and unconstrained.

Table 14: Metrics for different constraining methods applied to the SR CNN, calculated over the test samples of the NorESM data set. The mean is taken over 3 runs. Best scores are highlighted in bold.

DATA	MODEL	CONSTRAINT	RMSE	MAE	MS-SSIM	CONSTR. VIOL.
TAS NORESM	ENLARGE	NONE	2.987	1.915	95.96	**0.000**
TAS NORESM	BICUBIC	NONE	2.910	1.864	96.36	0.073
TAS NORESM	CNN	NONE	**2.348**	**1.559**	**96.93**	1.034
TAS NORESM	CNN	SOFT	2.928	1.874	96.28	0.041
TAS NORESM	CNN	ADDCL	2.885	1.847	96.45	**0.000**
TAS NORESM	CNN	SCADDCL	**2.884**	**1.846**	**96.46**	**0.000**
TAS NORESM	CNN	MULTCL	2.888	1.859	96.43	**0.000**
TAS NORESM	CNN	SMCL	2.885	1.847	96.45	**0.000**

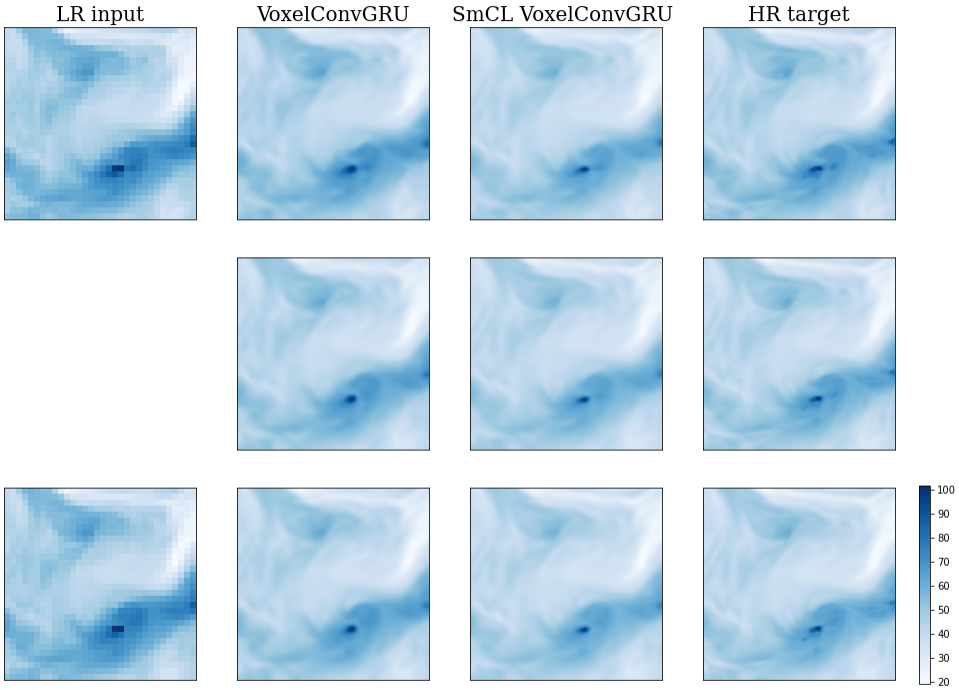

Figure 15: One random test sample and its prediction. Shown here are the two LR input time steps, predictions by both a constrained and unconstrained version of the FlowConvGRU, and the HR sequence as a reference.

Table 15: Metrics for different constraining methods applied to the SR-CNN, calculated over the test samples of the lunar data set. The mean is taken over 3 runs. The best scores are highlighted in bold blue.

Data	Model	Constraint	RMSE	MAE	SSIM	PSNR
Lunar	CNN	none	0.00217	0.00146	90.08	37.57
Lunar	CNN	SmCL	**0.00213**	**0.00144**	**90.40**	**37.74**

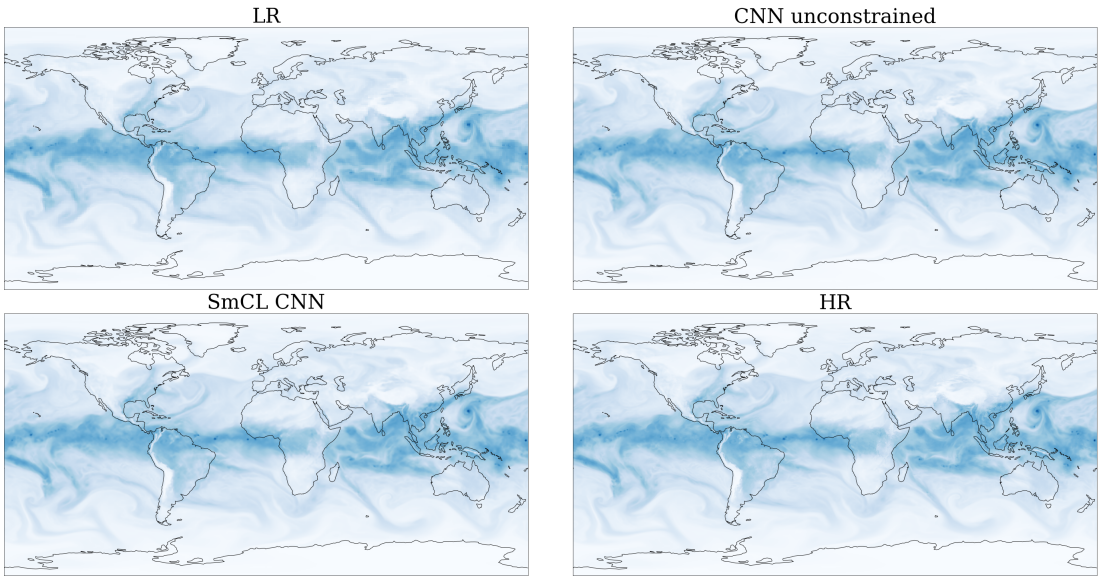

Figure 16: Global water content data (from data set TCW4): We show LR, unconstrained prediction, softmax-constrained prediction, and HR. The models are applied to one random time step of the test data set, seperately to each 32x32 patch and than combined together to create a global visualization.

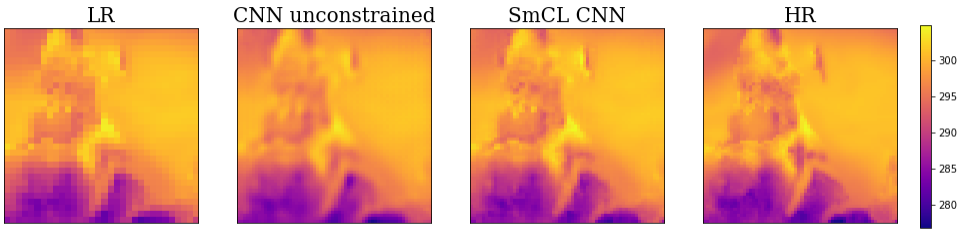

Figure 17: A random sample prediction for the NorESM temperature test data set, we compare an unconstrained CNN and a softmax-constrained CNN here. The constrained prediction looks more similar to the HR ground truth, including more high-frequency features.

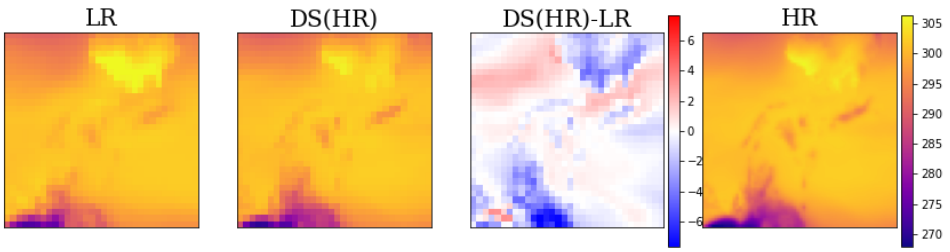

Figure 18: A sample from the NorESM temperature training data set. We compare the low-resolution simulation to the downsampled high-resolution counterpart. It can be observed that the LR and the downsampled HR are significantly different.

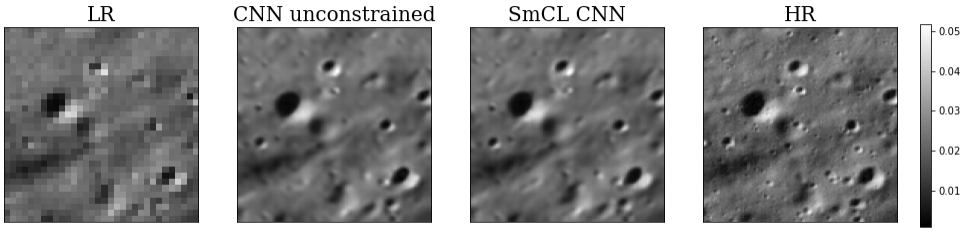

Figure 19: A random sample prediction from the lunar data set is shown. We compare the unconstrained with the constrained prediction.

Table 16: Metrics of the SR-GAN with and without SmCL calculated over the test data sets Set5, Set14, Urban100, BSD100. The better scores are highlighted in bold blue.

Data	Model	Constraint	RMSE	MAE	SSIM	PSNR
Set5	SR-GAN	none	8.57	4.80	92.48	29.47
Set5	SR-GAN	SmCL	**6.61**	**4.01**	**93.95**	**31.73**
Set14	SR-GAN	none	15.75	8.82	86.06	24.28
Set14	SR-GAN	SmCL	**14.07**	**8.12**	**87.37**	**25.16**
Urban100	SR-GAN	none	25.00	14.57	81.40	20.17
Urban100	SR-GAN	SmCL	**23.25**	**13.60**	**83.19**	**20.80**
BSD100	SR-GAN	none	14.38	8.28	85.95	24.97
BSD100	SR-GAN	SmCL	**13.52**	**7.82**	**87.09**	**25.50**

Figure 20: Two random images from both the BSD100 and the Urban100 data sets. The first row shows the unconstrained prediction, the second row the constrained prediction using softmax constraining.

References

G. A. R. Auger, C. D. Watson, and H. R. Kolar. The influence of weather forecast resolution on the circulation of lake george, ny. *Water Resources Research*, 57(10):e2020WR029552, 2021. doi: https://doi.org/10.1029/2020WR029552. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020WR029552. e2020WR029552 2020WR029552.

J. Baño Medina, R. Manzanas, and J. M. Gutiérrez. Configuration and intercomparison of deep learning neural models for statistical downscaling. *Geoscientific Model Development*, 13(4):2109–2124, 2020. doi: 10.5194/gmd-13-2109-2020. URL https://gmd.copernicus.org/articles/13/2109/2020/.

T. Beucler, S. Rasp, M. Pritchard, and P. Gentine. Achieving conservation of energy in neural network emulators for climate modeling, 2019. URL https://arxiv.org/abs/1906.06622.

T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine. Enforcing analytic constraints in neural networks emulating physical systems. *Phys. Rev. Lett.*, 126:098302, Mar 2021. doi: 10.1103/PhysRevLett.126.098302. URL https://link.aps.org/doi/10.1103/PhysRevLett.126.098302.

C. Chaudhuri and C. Robertson. Cligan: A structurally sensitive convolutional neural network model for statistical downscaling of precipitation from multi-model ensembles. *Water*, 2020.

J. Delgano-Centeno, P. Harder, B. Moseley, V. Bickel, S. Ganju, F. Kalaitzis, and M. Olivares-Mendez. Single image super-resolution with uncertainty estimation for lunar satellite images. *NeruIPS Workshop ML for Physical Sciences*, 2021.

C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016. doi: 10.1109/TPAMI.2015.2439281.

P. Donti, D. Rolnick, and J. Z. Kolter. Dc3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*, 2021.

A. Geiss and J. C. Hardin. Strictly enforcing invertibility and conservation in cnn-based super resolution for scientific datasets. *Artificial Intelligence for the Earth Systems*, 2(1): e210012, 2023. doi: https://doi.org/10.1175/AIES-D-21-0012.1. URL https://journals.ametsoc.org/view/journals/aies/2/1/AIES-D-21-0012.1.xml.

A. Geiss, S. Silva, and J. Hardin. Downscaling atmospheric chemistry simulations with physically consistent deep learning. *Geoscientific Model Development Discussions*, 2022: 1–26, 2022. doi: 10.5194/gmd-2022-76. URL https://gmd.copernicus.org/preprints/gmd-2022-76/.

B. Groenke, L. Madaus, and C. Monteleoni. Climalign: Unsupervised statistical downscaling of climate variables via normalizing flows. In *Proceedings of the 10th International Conference on Climate Informatics*, CI2020, page 60–66, New York, NY, USA, 2020.

Association for Computing Machinery. ISBN 9781450388481. doi: 10.1145/3429309.3429318. URL https://doi.org/10.1145/3429309.3429318.

W. J. Gutowski, P. A. Ullrich, A. Hall, L. R. Leung, T. A. O'Brien, C. M. Patricola, R. W. Arritt, M. S. Bukovsky, K. V. Calvin, Z. Feng, A. D. Jones, G. J. Kooperman, E. Monier, M. S. Pritchard, S. C. Pryor, Y. Qian, A. M. Rhoades, A. F. Roberts, K. Sakaguchi, N. Urban, and C. Zarzycki. The ongoing need for high-resolution regional climate models: Process understanding and stakeholder information. *Bulletin of the American Meteorological Society*, 101(5):E664 – E683, 2020. doi: 10.1175/BAMS-D-19-0113.1. URL https://journals.ametsoc.org/view/journals/bams/101/5/bams-d-19-0113.1.xml.

P. Harder, D. Watson-Parris, D. Strassel, N. Gauger, P. Stier, and J. Keuper. Physics-informed learning of aerosol microphysics. *arXiv preprint arXiv:2109.10593*, 2021.

P. Harder, D. Watson-Parris, P. Stier, D. Strassel, N. R. Gauger, and J. Keuper. Physics-informed learning of aerosol microphysics, 2022. URL https://arxiv.org/abs/2207.11786.

N. Harilal, M. Singh, and U. Bhatia. Augmented convolutional lstms for generation of high-resolution climate change projections. *IEEE Access*, 9:25208–25218, 2021. doi: 10.1109/ACCESS.2021.3057500.

H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. De Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flemming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R. J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. de Rosnay, I. Rozum, F. Vamborg, S. Villaume, and J.-N. Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020. doi: https://doi.org/10.1002/qj.3803. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803.

P. Hess, M. Drüke, S. Petri, F. M. Strnad, and N. Boers. Physically constrained generative adversarial networks for improving precipitation fields from earth system models. *Nature Machine Intelligence*, 4, 2022.

C. M. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, K. Kashinath, M. Mustafa, H. A. Tchelepi, P. Marcus, Prabhat, and A. Anandkumar. Meshfreeflownet: A physics-constrained deep continuous space-time super-resolution framework. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '20. IEEE Press, 2020. ISBN 9781728199986.

R. Kurinchi-Vendhan, B. Lütjens, R. Gupta, L. Werner, and D. Newman. Wisosuper: Benchmarking super-resolution methods on wind and solar data, 2021. URL https://arxiv.org/abs/2109.08770.

C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2016. URL https://arxiv.org/abs/1609.04802.

C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

J. Leinonen, D. Nerini, and A. Berne. Stochastic super-resolution for downscaling time-evolving atmospheric fields with a generative adversarial network. *IEEE Transactions on Geoscience and Remote Sensing*, 59(9):7211–7223, 2021. doi: 10.1109/TGRS.2020.3032790.

B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1132–1140, 2017. doi: 10.1109/CVPRW.2017.151.

Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4473–4481, 2017. doi: 10.1109/ICCV.2017.478.

A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte. Srflow: Learning the super-resolution space with normalizing flow. In *ECCV*, 2020.

D. Maraun and M. Widmann. *Statistical Downscaling and Bias Correction for Climate Research*. Cambridge University Press, 2018. doi: 10.1017/9781107588783.

M. Mirza and S. Osindero. Conditional generative adversarial nets, 2014. URL `https://arxiv.org/abs/1411.1784`.

J. Ott, M. Pritchard, N. Best, E. Linstead, M. Curcic, and P. Baldi. A fortran-keras deep learning bridge for scientific computing. *arXiv preprint arXiv:2004.10652*, 2020.

A. Quiquet, D. M. Roche, C. Dumas, and D. Paillard. Online dynamical downscaling of temperature and precipitation within the *i*loveclim model (version 1.1). *Geoscientific Model Development*, 11(1):453–466, 2018. doi: 10.5194/gmd-11-453-2018. URL `https://gmd.copernicus.org/articles/11/453/2018/`.

M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais, and Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 2 (1), 2019. doi: https://doi.org/10.1038/s41586-019-0912-1.

Ø. Seland, M. Bentsen, D. Olivié, T. Toniazzo, A. Gjermundsen, L. S. Graff, J. B. Debernard, A. K. Gupta, Y.-C. He, A. Kirkevåg, J. Schwinger, J. Tjiputra, K. S. Aas, I. Bethke, Y. Fan, J. Griesfeller, A. Grini, C. Guo, M. Ilicak, I. H. H. Karset, O. Landgren, J. Liakka, K. O. Moseid, A. Nummelin, C. Spensberger, H. Tang, Z. Zhang, C. Heinze, T. Iversen, and M. Schulz. Overview of the norwegian earth system model (noresm2) and key climate response of cmip6 deck, historical, and scenario simulations. *Geoscientific Model Development*, 13(12):6165–6200, 2020. doi: 10.5194/gmd-13-6165-2020. URL `https://gmd.copernicus.org/articles/13/6165/2020/`.

A. Serifi, T. Günther, and N. Ban. Spatio-temporal downscaling of climate data using convolutional and error-predicting neural networks. *Frontiers in Climate*, 3, 2021. ISSN 2624-9553. doi: 10.3389/fclim.2021.656479. URL https://www.frontiersin.org/articles/10.3389/fclim.2021.656479.

K. Stengel, A. Glaws, D. Hettinger, and R. N. King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117 (29):16805–16815, 2020. doi: 10.1073/pnas.1918964117. URL https://www.pnas.org/doi/abs/10.1073/pnas.1918964117.

T. Vandal, E. Kodra, S. Ganguly, A. Michaelis, R. Nemani, and A. R. Ganguly. Deepsd: Generating high resolution climate change projections through single image super-resolution. KDD '17, page 1663–1672, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450348874. doi: 10.1145/3097983.3098004. URL https://doi.org/10.1145/3097983.3098004.

J. Wang, Z. Liu, I. Foster, W. Chang, R. Kettimuthu, and V. R. Kotamarthi. Fast and accurate learned multiresolution dynamical downscaling for precipitation. *Geoscientific Model Development*, 14(10):6355–6372, 2021. doi: 10.5194/gmd-14-6355-2021. URL https://gmd.copernicus.org/articles/14/6355/2021/.

X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang. Esrgan: Enhanced super-resolution generative adversarial networks, 2018a. URL https://arxiv.org/abs/1809.00219.

X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *The European Conference on Computer Vision Workshops (ECCVW)*, September 2018b.

C. D. Watson, C. Wang, T. Lynar, and K. Weldemariam. Investigating two super-resolution methods for downscaling precipitation: ESRGAN and CAR, 2020. URL https://arxiv.org/abs/2012.01233.

F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo. Learning texture transformer network for image super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5790–5799, 2020. doi: 10.1109/CVPR42600.2020.00583.

L. Zanna and T. Bolton. Data-driven equation discovery of ocean mesoscale closures. *Geophysical Research Letters*, 47(17):e2020GL088376, 2020. doi: https://doi.org/10.1029/2020GL088376. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2020GL088376. e2020GL088376 10.1029/2020GL088376.

L. Zanna and T. Bolton. *Deep Learning of Unresolved Turbulent Ocean Processes in Climate Models*, chapter 20, pages 298–306. John Wiley & Sons, Ltd, 2021. ISBN 9781119646181. doi: https://doi.org/10.1002/9781119646181.ch20. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119646181.ch20.