Generating physically-consistent high-resolution climate data with hard-constrained neural networks

Paula Harder,[1, 2, 3] **Qidong Yang,** [1, 4] **Venkatesh Ramesh,** [1, 5] **Alex Hernandez-Garcia,** [1, 5] **Prasanna Sattigeri,** [6] **Campbell D. Watson,** [6] **Daniela Szwarcman,**[6] **David Rolnick** [1, 7]

[1] Mila Quebec AI Institute
[2] Fraunhofer ITWM
[3] TU Kaiserslautern
[4] New York University
[5] University of Montreal
[6] IBM Research
[7] McGill University
paula.harder@mila.quebec

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 can only make coarse resolution predictions. Statistical downscaling can provide an efficient method of upsampling low-resolution data. In this field, deep learning has been applied successfully, often using image super-resolution methods from computer vision. Despite achieving visually compelling results in some cases, such models often violate conservation laws when predicting physical variables. In order to conserve important physical quantities, we develop methods that guarantee physical constraints are satisfied by a deep downscaling model while also increasing their performance according to traditional metrics. We introduce two ways of constraining the network: A renormalization layer added to the end of the neural network and a successive approach that scales with increasing upsampling factors. We show the applicability of our methods across different popular architectures and upsampling factors using ERA5 reanalysis data.

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 shape adaptation decisions and provide impetus for action to reduce greenhouse gas emissions. Predicted and observed quantities such as precipitation, wind speed, and temperature impact decisions in fields like 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, resulting in coarse-resolution predictions, while observations are often sparsely available. 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 learning a mapping between a low-resolution (LR) input sample and a high-resolution (HR) version of that sample. 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 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. 2018), 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 spatio-temporal climate data. Recently, deep learning approaches have been applied to a variety of climate and weather datasets, covering both model output data and observations. Climate super-resolution has mostly focused on CNNs (Vandal et al. 2017), recently shifting towards GANs (Stengel et al. 2020).

Generating high-resolution data with machine learning can produce realistic-looking images and good predictive accuracy. However, a major obstacle often encountered when applying ML to a physical system such as the Earth's atmosphere is that the predicted output values can violate physical laws such as conservation of energy, momentum, and moisture. More broadly, there are numerous domains of ML for societal benefit in which satisfaction of physical constraints is fundamentally important. Examples include the discovery of new materials for energy and healthcare, aerodynamics simulations for efficient vehicles, and optimal control in industrial settings.

In this work, we introduce a novel method to strictly enforce physical constraints between low-resolution (input) and high-resolution (output) images. Using ERA5 reanalysis data, we build datasets to learn the upsampling with differ-

ent factors ranging from 2 up to 16 times enhancement. We introduce a renormalization layer that can either be added as the last layer to any network architecture or successively after each upsampling part, developing a new SR network architecture. Besides looking at spatial SR, we also include temporal SR or frame interpolation, demonstrating that our methodology works across all these architectures.

Our novel constraining methodology is not only applicable to SR tasks in scientific areas, but it can also constrain neural networks for emulation and prediction. Climate model emulation tasks like done by Beucler et al. (2021) and Harder et al. (2021) could benefit from using our constraining layer. Within the SR domain, our work could have implications beyond scientific datasets as the constraining methodologies can advance state-of-the-art SR more generally.

Contributions Our main contributions can be summarized as follows:

- We introduce a novel constraining methodology and show that it improves the performance of a wide variety of deep learning architectures for super-resolution. In general, it can be applied to any existing neural architecture.
- Applied to a super-resolution neural network our constraining methodologies guarantee that physical constraints such as mass conservation are satisfied in the prediction.
- Besides a single constraints layer we introduce a successive constraining methodology that makes constrained super-resolution more feasible for large upscaling factors.
- We introduce a new deep learning architecture for downscaling that both increases the spatial and temporal resolution.

Related Work

Deep Learning for Climate Downscaling There exists extensive work on ML methods for climate and weather observations and predictions, from CNN architectures (Vandal et al. 2017) to GANs (Stengel et al. 2020) and normalizing flows (Groenke, Madaus, and Monteleoni 2020). Recently, especially 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). First unified frameworks comparing methods and benchmarks were introduced: Baño Medina, Manzanas, and Gutiérrez (2020) assess different SR-CNN setups and Kurinchi-Vendhan et al. (2021) published a dataset for wind and solar SR data. An area with only limited research to this date—despite its high practical relevance—is performing spatio-temporal SR. Some papers have looked at super-resolving multiple time steps at once, but not increasing the temporal resolution (Harilal, Singh, and Bhatia 2021; Leinonen, Nerini, and Berne 2021), whereas Serifi, Günther, and Ban (2021) increases the temporal resolution by just 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) or linearly constrained neural networks for convection (Beucler et al. 2021) or aerosol microphysics emulation (Harder et al. 2022). Outside climate science, recent works have emerged on enforcing hard constraints on the output of neural networks (Donti, Rolnick, and Kolter 2021).

Constrained Learning for Downscaling For solving super-resolution for turbulent flows, MeshfreeFlowNet (Jiang et al. 2020) employs a physics-informed model adding PDEs as regularization terms to the loss function. To the best of our knowledge, there is no published work on hard-constrained ML models for SR/downscaling. A preprint (Geiss and Hardin 2020) introduces an enforcement operator applied to multiple CNN architectures for scientific datasets while another more recent preprint applies that method to atmospheric chemistry model downscaling using a CNN (Geiss, Silva, and Hardin 2022).

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, while 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).

Setup

Consider the case of N times downscaling and let $n := N^2$. Let $y_i, i = 1, \ldots, n$ be the values in the predicted high-resolution data point that correspond to low-resolution pixel x. The downscaling/conservation constraint is given by the following:

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

As we model mass per area, conserving the mean in the SR image means conserving the overall mass.

Constraints Layer

Let $\tilde{y}_i, i = 1, \ldots, n$ be the intermediate outputs of the neural network before the constraints layer. At the end of a network architecture we take these outputs and rescale them using the corresponding input value x:

$$y_j = \tilde{y}_j \cdot \frac{x}{\frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i}. \tag{2}$$

This operation is called the constraints layer (CL), it guarantees Equation 2 to be satisfied and it can be added to the end of a neural network as shown in Figure 2.

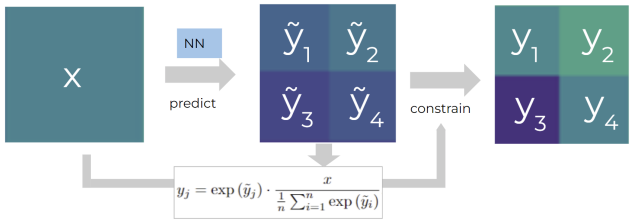

Figure 1: Our constraining methodology shown for one input pixel x and the softmax constraining on the corresponding predicted 2×2 patch for the case of 2 times upsampling.

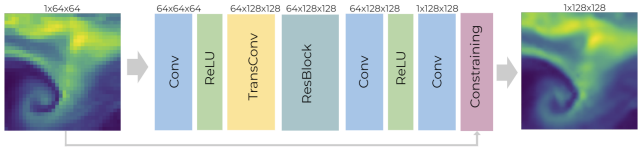

Figure 2: The CNN architecture for 2 times upsampling including the constraints layer (in red). For higher upsampling factors this architecture is applied recursively, either with (see successive constraining) or without the constraints layer.

For predicting quantities like atmospheric water content, we also want to enforce the output to be positive and, therefore, physically valid. Here, we use a softmax multiplied by the corresponding input pixel value x:

$$y_j = \exp(\tilde{y}_j) \cdot \frac{x}{\frac{1}{n}\sum_{i=1}^{n} \exp(\tilde{y}_i)}. \quad (3)$$

We call this layer the softmax constraints layer (SMCL). It not only enforces our equality constraints, but also our positivity constraints: $y_i \geq 0, i = 1, \ldots, n$. Including the exponential function also changes the distribution, making it more extreme/peakier, which can help with the common problem of having too smooth predictions while optimizing the MSE.

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.

Successive Constraining

Building on the constraints layer introduced before, we develop a methodology that scales well with increasing upsampling factors. Here the constraints layer is applied multiple times, increasing the resolution by a factor of 2 each time, via either our CL or SMCL. An example is shown in Figure 3: Starting from our LR input, a medium-resolution intermediate prediction is produced, which is then constrained using the LR image. The final SR image is constrained using the medium-resolution one.

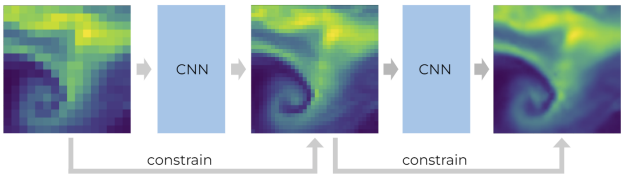

Figure 3: Our successive constraining approach visualized for 4 times upsampling here. Each of the CNNs is built as shown in Figure 2. A medium-resolution intermediate output is predicted that is constrained by the LR image and then used for constraining the final SR prediction.

Experimental Setup

Data

ERA5 Dataset The ERA5 dataset is a so-called *reanalysis* product from the European Center for Medium-Range Weather Forecast (ECMWF) that combines model data with worldwide observations. The observations are used as boundary conditions for numerical models that then predict various atmospheric variables. ERA5 is available as global, hourly data with a $0.25° \times 0.25°$ resolution, which is roughly 25 km per pixel. It covers all years starting from 1950. For this work, we focus on the total column water (tcw) that is given in $\frac{\text{kg}}{\text{m}^2}$ and describes the vertical integral of the total amount of atmospheric water content, including water vapour, cloud water, and cloud ice but not precipitation.

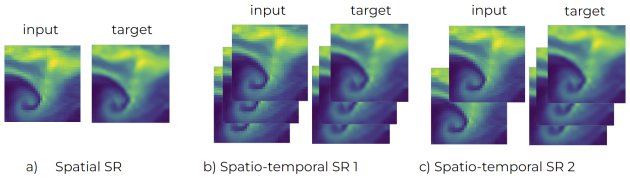

Figure 4: Samples of the three different dataset types used in this work. a) A data pair we use for our standard spatial super-resolution task. The input is an LR image and the target is the HR version of that. b) A data pair for performing SR for multiple time steps simultaneously. The input is a time 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.

Spatial SR Dataset 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 mean over $N \times N$ patches, where N is our upsampling factor, following standard practice as in e.g. Serifi, Günther, and Ban (2021). A sample pair is shown in Figure 4 a). This is the physically correct way to create the

low-resolution images since conservation of water content means that the water content (density per squared meter) described in an LR pixel should equal the mean of the values in the corresponding HR pixels.

Spatio-Temporal Datasets Including the temporal evolution of our data, we create two additional datasets. For the first dataset, 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 4 b). Per global image we select 3 random 128×128 pixel areas, 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. To increase both spatial and temporal dimensions, we again crop 3 images out of a series of 3 successive time steps. To create the low-resolution input, we take every other time step and compute the mean spatially, resulting in 2 LR inputs, see Figure 4 c).

Architectures

We test our constraints 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 as well as apply them to 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 serve as a comparison for our constraining methodologies.

SR-CNNs The SR CNN network consists of convolutional layers using 3×3 kernels and ReLU activations. The upsampling is done by a transpose convolution followed by residual blocks. The architecture for 2 times downscaling is shown in Figure 2.

SR-GAN The conditional GAN architecture uses the above introduced CNN architecture as the generator network. The discriminator simply consists of convolutional layers with a stride of 2 to decrease the dimensionality in each step and ReLU activations. It is trained as a classifier to distinguish SR images from real HR images using a binary cross entropy loss. The conditional GAN 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 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 presented by Leinonen, Nerini, and Berne (2021), which uses ConvGRU layers to address the spatio-temporal nature of super-resolving a time-series of climate data.

SR-VoxelConvGRU To increase the temporal resolution of our data we employ the Deep Voxel 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 by stacking the Deep Voxel Flow model and the ConvGRU network (VoxelConvGRU): 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.

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.

Baselines

As a simple non-ML baseline we use bicubic interpolation for spatial SR and take the mean of two frames for temporal SR. As a stronger baseline, we also run experiments with the enforcement operator, introduced by Geiss and Hardin (2020), which we refer to as *G-H*. Their constraining operator, tested in the context of SR CNNs only, is as follows:

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

with $\sigma := \text{sign}(\frac{1}{n}\sum_{i=1}^{n} \tilde{y}_i - x)$. Furthermore, we always compare against an unconstrained version of the above introduced standard SR NN architectures (SR-CNN, SR-GAN, SR-ConvGRU, SR-VoxelConvGRU).

Results

As standard SR metrics, we report the peak signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) for all our experiments. Additionally, we report how much mass conservation is violated.

We can see in Tables 1, 2, 3 and 4 that for all cases applying constraining methodologies does not only enforce mass conservation, but also improves both PSNR and SSIM scores. This shows that we achieve better performance than standard SR neural architectures without a constraining layer. Due to numerical reasons, the mass conservation scores vary across constraining methodologies. The G-H operator is in all cases weaker when it comes to exactly enforcing the constraints compared to our constraining methods, which achieve up to an order of magnitude more accurate mass conservations. Overall the predicted values have a mean value of around $20 \frac{kg}{m^2}$, therefore the violations are relatively small among all constraining methodologies. For each upsampling factor and both CNNs and GANs applying constraints can increase PSNR and SSIM scores compared to the same architecture without any constraints. In general, the SMCL is giving improved results in comparison with the CL. Successive constraining shows slightly better results than a single constraints layer but the achieved improvement decreases with bigger upsampling factors. For GANs the improvement caused by constraining is even more pronounced than in CNNs. An example of spatial SR prediction for different methods can be found in Figure 5. Tables 3 and 4 show that constraining works well also for spatio-temporal downscaling. Especially, our VoxelConvGRU architecture bene-

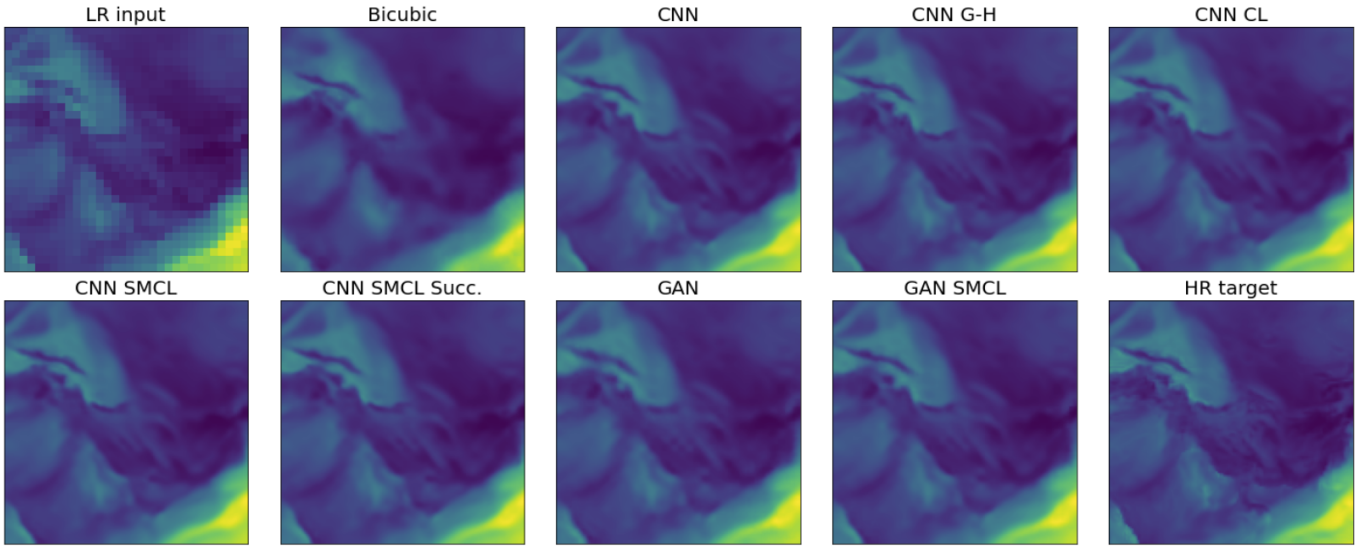

Figure 5: One example image chosen randomly from the test set. Shown here is the LR input, different constrained and unconstrained predictions and the HR image as a reference.

Table 1: Metrics for different constraining methods applied to an SR CNN, calculated over 10,000 test samples. Best scores are highlighted in bold, second best in blue. The metrics shown are PSNR/SSIM/Mass violation.

Model	Constraint	2x	4x	8x	16x
Bicubic	-	$52.2/99.88/6.6e^{-2}$	$44.3/99.29/1.7e^{-1}$	$38.5/98.08/3.2e^{-1}$	$33.6/96.62/5.1e^{-1}$
SR-CNN	none	$56.0/99.94/1.5e^{-2}$	$47.5/99.58/3.1e^{-2}$	$40.7/98.56/6.0e^{-2}$	$35.1/\mathbf{97.1}/8.3e^{-2}$
SR-CNN	G-H	$\mathbf{57.0}/\mathbf{99.95}/1.6e^{-6}$	$\mathbf{47.9}/99.60/1.1e^{-5}$	$40.8/98.55/4.4e^{-6}$	$35.1/97.0/1.7e^{-6}$
SR-CNN	CL (ours)	$56.3/99.94/\mathbf{8.6e^{-7}}$	$\mathbf{47.9}/99.60/1.3e^{-6}$	$40.5/98.48/1.2e^{-6}$	$35.1/97.0/4.4e^{-6}$
SR-CNN	SMCL (ours)	$\mathbf{57.0}/\mathbf{99.95}/8.7e^{-7}$	$\mathbf{47.9}/99.60/1.4e^{-6}$	$40.7/98.54/2.2e^{-6}$	$35.1/\mathbf{97.1}/4.4e^{-6}$
SR-CNN	CL Succ. (ours)	-	$47.8/99.60/\mathbf{1.1e^{-6}}$	$40.8/98.57/6.2e^{-7}$	$\mathbf{35.2}/\mathbf{97.1}/\mathbf{7.0e^{-7}}$
SR-CNN	SMCL Succ. (ours)	-	$\mathbf{47.9}/\mathbf{99.61}/\mathbf{1.1e^{-6}}$	$\mathbf{40.9}/\mathbf{98.59}/\mathbf{6.0e^{-7}}$	$\mathbf{35.2}/\mathbf{97.1}/7.1e^{-7}$

Table 2: Scores for different constraining methods applied to a SR GAN architecture, calculated over 10,000 test samples. Best scores are highlighted in bold, second best in blue. The metrics shown are PSNR/SSIM/Mass violation.

Model	Constraint	Metrics
Bicubic	-	$44.3/99.29/1.7e^{-1}$
SR-GAN	none	$46.5/99.50/4.6e^{-2}$
SR-GAN	G-H	$\mathbf{47.4}/\mathbf{99.56}/2.3e^{-6}$
SR-GAN	CL (ours)	$46.9/99.53/1.4e^{-6}$
SR-GAN	SMCL (ours)	$47.3/\mathbf{99.56}/\mathbf{1.3e^{-6}}$

Table 3: Scores for different constraining methods applied to the SR ConvGRU architecture, calculated over 10,000 test samples for 4 times upsampling. Best scores are highlighted in bold, second best in blue. The metrics shown are PSNR/SSIM/Mass violation.

Model	Constraint	Metrics
Bicubic	-	$44.2/99.28/1.7e^{-1}$
SR-ConvGRU	none	$44.8/99.37/1.8e^{-1}$
SR-ConvGRU	G-H	$\mathbf{48.3}/\mathbf{99.64}/5.5e^{-5}$
SR-ConvGRU	CL (ours)	$44.7/99.30/\mathbf{1.3e^{-6}}$
SR-ConvGRU	SMCL (ours)	$48.2/99.63/\mathbf{1.3e^{-6}}$

fits from the SCML methods. A time-series plot is shown in Figure 7.

Conclusion

This work shows a novel methodology to incorporate physics-based constraints into neural network architectures for climate downscaling. Our constrained models not only guarantee to satisfy conservation laws like mass conservation or their outputs but also increase predictive performance across different architectures. Additionally, we introduce a new architecture that does both super-resolution in spatial and temporal domain, while still respecting mass conservation. We present the first approach to progressively down-

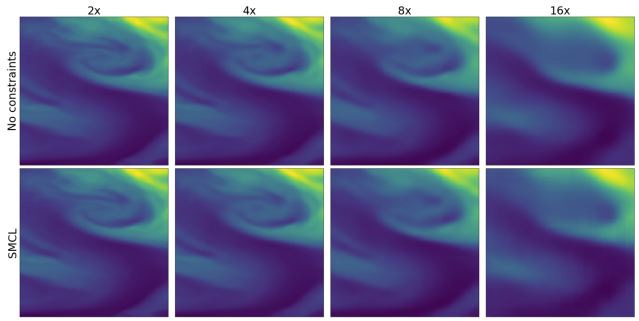

Figure 6: One example image chosen randomly from the test set. Each model was trained for the same target resolution but with a different upsampling factor. First row shows the prediction of an unconstrained CNN, second row one that leverages the SMCL.

Table 4: Scores for different constraining methods applied to our VoxelConvGRU SR architecture, calculated over 10,000 test samples for 4 times upsampling spatially and 2 times temporally. Best scores are highlighted in bold, second best in blue.

MODEL	CONSTRAINT	METRICS
BICUBIC+INTERP.	-	$42.9/99.18/3.1e^{-1}$
SR-VOXELCONVGRU	NONE	$45.1/99.40/1.5e^{-1}$
SR-VOXELCONVGRU	G-H	$44.7/99.38/1.7e^{-6}$
SR-VOXELCONVGRU	CL (OURS)	$44.0/99.16/\mathbf{8.7e^{-7}}$
SR-VOXELCONVGRU	SMCL (OURS)	$\mathbf{47.9/99.62/8.7e^{-7}}$

scale and constrain, introducing a promising method for large-scale downscaling.

Next steps include extending the evaluation to different SR climate datasets as well as applying the constraining layers to other climate or scientific ML tasks. One possible future direction could be transferring the constraining methodologies to super-resolution task in general and advancing their performance using our methods.

References

Baño Medina, J.; Manzanas, R.; and Gutiérrez, J. M. 2020. Configuration and intercomparison of deep learning neural models for statistical downscaling. *Geoscientific Model Development*, 13(4): 2109–2124.

Beucler, T.; Pritchard, M.; Rasp, S.; Ott, J.; Baldi, P.; and Gentine, P. 2021. Enforcing Analytic Constraints in Neural Networks Emulating Physical Systems. *Phys. Rev. Lett.*, 126: 098302.

Beucler, T.; Rasp, S.; Pritchard, M.; and Gentine, P. 2019. Achieving Conservation of Energy in Neural Network Emulators for Climate Modeling.

Chaudhuri, C.; and Robertson, C. 2020. CliGAN: A Structurally Sensitive Convolutional Neural Network Model for Statistical Downscaling of Precipitation from Multi-Model Ensembles. *Water*.

Dong, C.; Loy, C. C.; He, K.; and Tang, X. 2016. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2): 295–307.

Donti, P.; Rolnick, D.; and Kolter, J. Z. 2021. DC3: A learning method for optimization with hard constraints. In *International Conference on Learning Representations*.

Geiss, A.; and Hardin, J. C. 2020. Strict Enforcement of Conservation Laws and Invertibility in CNN-Based Super Resolution for Scientific Datasets.

Geiss, A.; Silva, S.; and Hardin, J. 2022. Downscaling Atmospheric Chemistry Simulations with Physically Consistent Deep Learning. *Geoscientific Model Development Discussions*, 2022: 1–26.

Groenke, B.; Madaus, L.; and Monteleoni, C. 2020. ClimAlign: Unsupervised Statistical Downscaling of Climate Variables via Normalizing Flows. In *Proceedings of the 10th International Conference on Climate Informatics*, CI2020, 60–66. New York, NY, USA: Association for Computing Machinery. ISBN 9781450388481.

Harder, P.; Watson-Parris, D.; Stier, P.; Strassel, D.; Gauger, N. R.; and Keuper, J. 2022. Physics-Informed Learning of Aerosol Microphysics.

Harder, P.; Watson-Parris, D.; Strassel, D.; Gauger, N.; Stier, P.; and Keuper, J. 2021. Physics-Informed Learning of Aerosol Microphysics. *arXiv preprint arXiv:2109.10593*.

Harilal, N.; Singh, M.; and Bhatia, U. 2021. Augmented Convolutional LSTMs for Generation of High-Resolution Climate Change Projections. *IEEE Access*, 9: 25208–25218.

Jiang, C. M.; Esmaeilzadeh, S.; Azizzadenesheli, K.; Kashinath, K.; Mustafa, M.; Tchelepi, H. A.; Marcus, P.; Prabhat; and Anandkumar, A. 2020. 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. ISBN 9781728199986.

Kurinchi-Vendhan, R.; Lütjens, B.; Gupta, R.; Werner, L.; and Newman, D. 2021. WiSoSuper: Benchmarking Super-Resolution Methods on Wind and Solar Data.

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

Leinonen, J.; Nerini, D.; and Berne, A. 2021. 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.

Liu, Z.; Yeh, R. A.; Tang, X.; Liu, Y.; and Agarwala, A. 2017. Video Frame Synthesis Using Deep Voxel Flow. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 4473–4481.

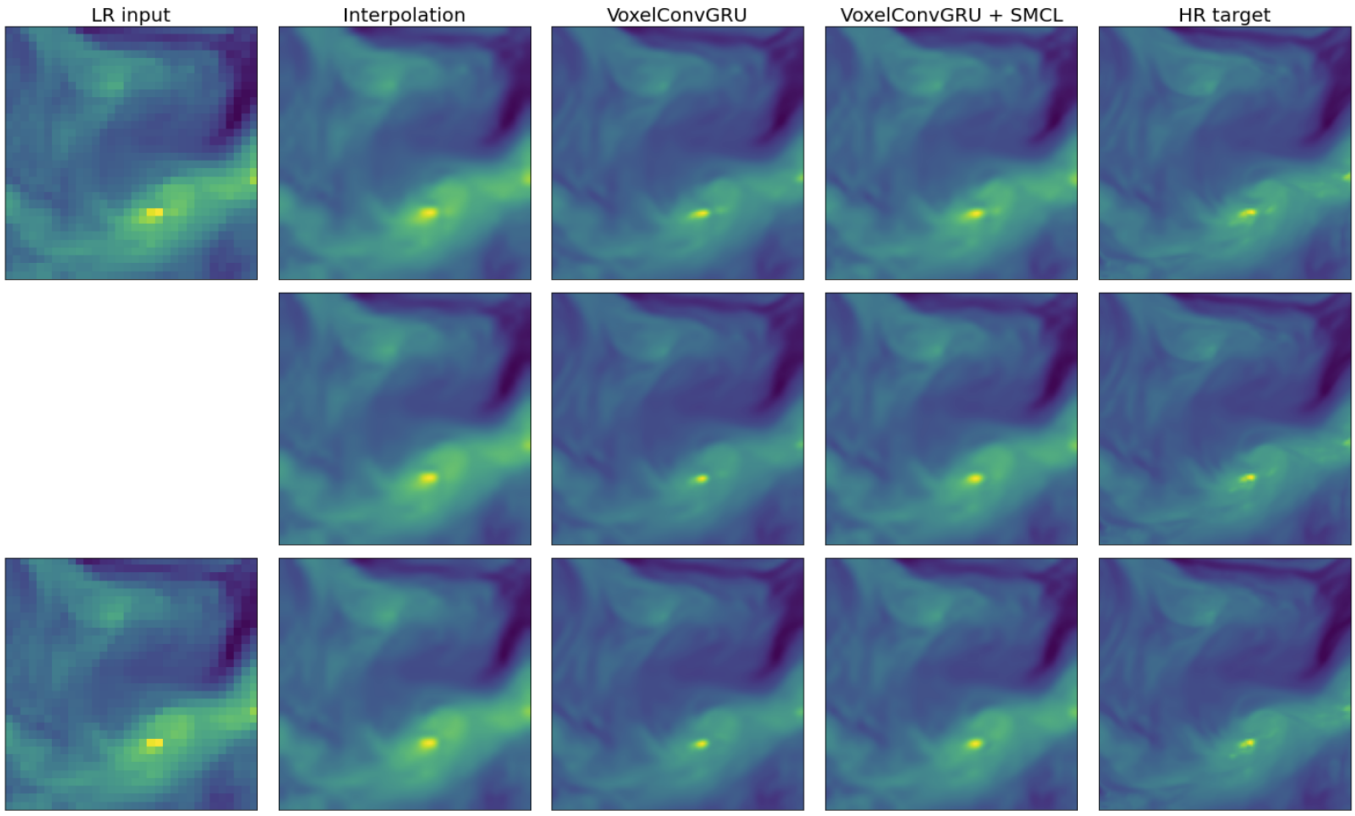

Figure 7: One random test sample and it's prediction. Shown here are the two LR input time steps, predictions by both a constrained and unconstrained version of the VoxelConvGRU, and the HR sequence as a reference.

Lugmayr, A.; Danelljan, M.; Van Gool, L.; and Timofte, R. 2020. SRFlow: Learning the Super-Resolution Space with Normalizing Flow. In *ECCV*.

Maraun, D.; and Widmann, M. 2018. *Statistical Downscaling and Bias Correction for Climate Research*. Cambridge University Press.

Serifi, A.; Günther, T.; and Ban, N. 2021. Spatio-Temporal Downscaling of Climate Data Using Convolutional and Error-Predicting Neural Networks. *Frontiers in Climate*, 3.

Stengel, K.; Glaws, A.; Hettinger, D.; and King, R. N. 2020. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29): 16805–16815.

Vandal, T.; Kodra, E.; Ganguly, S.; Michaelis, A.; Nemani, R.; and Ganguly, A. R. 2017. DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution. KDD '17, 1663–1672. New York, NY, USA: Association for Computing Machinery. ISBN 9781450348874.

Wang, J.; Liu, Z.; Foster, I.; Chang, W.; Kettimuthu, R.; and Kotamarthi, V. R. 2021. Fast and accurate learned multiresolution dynamical downscaling for precipitation. *Geoscientific Model Development*, 14(10): 6355–6372.

Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Loy, C. C.; Qiao, Y.; and Tang, X. 2018. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks.

Watson, C. D.; Wang, C.; Lynar, T.; and Weldemariam, K. 2020. Investigating two super-resolution methods for downscaling precipitation: ESRGAN and CAR.

Yang, F.; Yang, H.; Fu, J.; Lu, H.; and Guo, B. 2020. Learning Texture Transformer Network for Image Super-Resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5790–5799.