Vision, Modeling, and Visualization (2022)
J. Bender, M. Botsch, and D. Keim (Eds.)

Evaluation of Volume Representation Networks for Meteorological Ensemble Compression

K. Höhlein[1], S. Weiss[1] and R. Westermann[1]

[1]Technical University of Munich, Department of Informatics, Germany

Abstract

Recent studies have shown that volume scene representation networks constitute powerful means to transform 3D scalar fields into extremely compact representations, from which the initial field samples can be randomly accessed. In this work, we evaluate the capabilities of such networks to compress meteorological ensemble data, which are comprised of many separate weather forecast simulations. We analyze whether these networks can effectively exploit similarities between the ensemble members, and how alternative classical compression approaches perform in comparison. Since meteorological ensembles contain different physical parameters with various statistical characteristics and variations on multiple scales of magnitude, we analyze the impact of data normalization schemes on learning quality. Along with an evaluation of the trade-offs between reconstruction quality and network model parameterization, we compare compression ratios and reconstruction quality for different model architectures and alternative compression schemes.

CCS Concepts

• ***Computing methodologies*** → ***Learning latent representations;*** • ***Applied computing*** → ***Earth and atmospheric sciences;***

1. Introduction

Meteorological ensemble data comprise multiple weather forecast simulations, which can differ in initial conditions, numerical approximations or even physical model assumptions, and are used to assess uncertainties of the forecast outcome. Over the last decade, researchers have continually pushed ensemble sizes to larger scales, while, at the same time, extending spatial domain size, resolution and time horizon. Thus, meteorological ensembles can become extremely large. Ensembles are produced daily by weather centers and require large amounts of secondary disk space for backup.

Due to the shear volume of meteorological ensemble data, any attempt to analyse such datasets is intrinsically difficult. In the scenario we consider, the ensemble dataset comprises 1000 runs of a high-resolution numerical atmospheric dynamics model [NGW*20], thus pushing the data volume to 60 GB of memory for only a single time step. This makes it impossible to keep the data entirely on recent GPUs and fosters the need for effective compression schemes for multi-dimensional arrays of floating-point data. Yet, besides targeted strategies for reducing I/O bandwidth and storage requirements by converting such ensembles into compact data representations, random access to the data is mandatory to avoid decoding the entire ensemble for analysis tasks.

While lossless compression schemes allow for bit-wise accurate reconstruction of the original data, they typically achieve up to only 2x compression or less [SCH*14]. Lossy data compression schemes, such as ZFP [Lin14], SZ [DC16], or TThresh [BRLP19], in contrast, offer higher compression ratios of 100x or more, at the cost of introducing noticeable reconstruction errors. For most downstream analysis tasks, however, a certain error level is acceptable, such that lossy compression becomes a suitable tool for memory reduction [BHM*16, CDL*19].

As an alternative to classical lossy compressors for multi-dimensional scalar fields, compression schemes based on fully-connected neural networks have been proposed recently. Volume scene representation networks (V-SRNs) have been introduced by Lu *et al.* [LJLB21], and were further improved and accelerated by Weiss *et al.* [WHW21] (fV-SRN). V-SRNs are an extension of scene representation networks (SRNs), which were first developed for representing opaque surface models [MON*19,CZ19,PFS*19]. Besides offering the ability to directly reconstruct single samples from the compressed representation, V-SRNs are capable of exploiting non-local coherence in the data [CLI*20]. This makes V-SRNs a promising tool for compressing meteorological ensemble data, in which coherence and correlation are often observed between multiple parameter fields of the same simulation run or between different members of the same ensemble, but are more difficult to exploit for compression than, e.g., auto-correlations in space and time.

Contribution In this work, we evaluate the potential of V-SRNs for learning compact representations of ensembles of volumetric multi-parameter fields. We compare two different model architec-

DOI: 10.2312/vmv.20221198

tures, which allow for efficient parameter sharing between multiple parameter fields and ensemble members. We demonstrate that this results in compression rates that are higher or on par with those achieved by classical compression schemes, which have been adapted to exploit redundancy between different ensemble members. We do not focus on data with temporal variability explicitly, but our methods generalize straight-forwardly also to ensembles of time-variate multi-parameter fields.

We propose and analyze binary model architectures, leveraging a combination of low-resolution grids of trainable spatial latent features and small neural networks to read out the features and serve as non-linear interpolation functions. Different combinations of grids and networks are evaluated to identify trade-offs between model parameterization, reconstruction accuracy, and compression rate. Our analyses are tightly coupled to a case study using meteorological simulation data from a convective-scale ensemble by Necker *et al.* [NGW*20]. Based on this ensemble dataset, we discuss general methodological aspects, such as model design and training procedures, and highlight the importance of data-related aspects, such as the impact of data normalization. The code for the project is publically available at [HW22].

2. Related Work

Scene representation networks The concept of scene representation networks (SRNs) was concurrently introduced by Mescheder *et al.* [MON*19], Chen and Zhang [CZ19] and Park *et al.* [PFS*19], who present the idea of encoding an opaque, uncolored surface model as an implicit function that is implemented as a fully-connected neural network. The authors use feature vectors to encode object specific information and enable reusing models for different objects. The idea of trainable latent features was developed further by Chabra *et al.* [CLI*20], who replace the single feature vector by a feature grid to improve reconstruction accuracy. Multiple studies explore improvements and extensions of this idea. Martel *et al.* [MLL*21] use an adaptive data structure that is refined during training to allocate more resources in areas of larger errors. A fixed multi-resolution grid is used by Takikawa *et al.* [TLY*21] and later extended with spatial hashing by Müller *et al.* [MESK22], together with an efficient network implementation [MRNK21]. For a more comprehensive review of SRN-related literature, we refer to the overview articles by Hoang *et al.* [HSB*20] and Tewari *et al.* [TFT*20]. The works by Lu *et al.* [LJLB21] and Weiss *et al.* [WHW21] extend SRNs for volumetric data compression. The latter contributes in particular a fast network evaluation method to speed up training and decompression. Mishra *et al.* [MHBB22] leverage fully-connected neural networks for interpolating scientific data. We build upon and extend these works by focusing explicitly on the multi-parameter and ensemble compression capabilities of V-SRNs.

Lossy volume compression schemes Prior work in the area of lossy compression schemes can be categorized into three classes of algorithms. Transform coding-based schemes [YL95, LCA08] employ the discrete cosine or wavelet transformation to transform the data into a basis in which only few coefficients are relevant, while many others can be removed. Quantization schemes represent contiguous data blocks by a single index or a sparse combination of learned representative values [SW03, FM07, GIGM12, GG16]. One instance of this class of compression algorithms is the SZ algorithm [DC16, ZDL*20] using lossy curve fittings. Tensor decomposition schemes decompose the data directly using, e.g., a singular value decomposition. As one instance of such schemes, TThresh [BRLP19] can achieve extremely high compression ratios of 1000x or more. In interactive scenarios, mostly transform coding-based schemes are applied brick-wise, in which case high compression ratios are traded in on fast GPU-based decompression, see e.g. [DMG20, MAG19]. Focusing on applied scientific data compression, various studies have evaluated the applicability and performance of lossless and lossy data compression algorithms on atmospheric datasets [HWK*13, BHM*16, DCG19, KRD*21], and Dueben *et al.* [DLB19] discuss methods for efficient storage of weather forecast ensembles. Baker *et al.* [BPH22] have introduced a data-based similarity measure, termed DSSIM, for evaluating the quality loss in scientific data after lossy compression.

Table 1: *List of available simulation parameters.*

Name (Short name)	Unit	Value range
Temperature (tk)	Kelvin	$[200, 300]$
3D wind speed (u, v, w)	ms^{-1}	$[-40, 40]$
relative humidity (rh)	%	$[0, 100]$
water vapor mixing ratio (qv)	1	$[0, 0.02]$
mixing ratio of hydrometers (qhydro)	1	$[0, 0.01]$
geopotential height (z)	m	$[200, 20000]$
radar reflectivity (dbz)	dBZ	$[-30, 40]$

3. Data

We evaluate the compression capabilities of V-SRNs on a multi-parameter ensemble dataset, which was generated to study correlation patterns in atmospheric dynamics [NGW*20]. The dataset comprises 1000 runs of an atmospheric dynamics model over a rectangular domain in central Europe. Nine prognostic atmospheric parameters are stored at regular time steps of one hour, on a rectangular grid with 352×250 nodes and 20 levels in height. Due to the presence of mountain ranges and topography in the simulated domain, large parts of the data in lower levels are missing due to grid cells lying below the earth surface. For simplicity, we omit grid levels with missing values and restrict the dataset to the 12 top-most levels, which are free of missing values. A list of the available parameters is given in Tab. 1. The fields possess different physical interpretations and differ in value range and statistical distribution. As shown in Fig. 1, the distribution of field values varies not only between different parameters, but also between height levels of the same field, which complicates the learning task for deep learning-based compression algorithms.

Data normalization To facilitate model optimization, we examine the effect of different normalization methods, which rescale all parameters to a value range of $[0, 1]$. In the context of data compression, data rescaling has been discussed by Dueben *et al.* [DLB19] and was found to improve compression efficiency. We compare

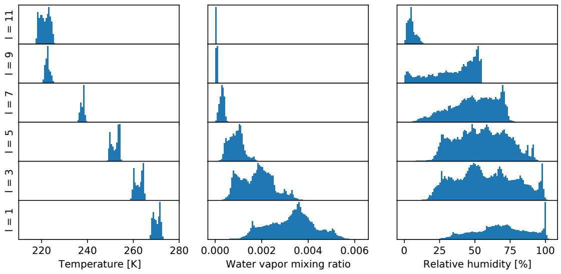

Figure 1: *Value distribution marginalized over different height levels for parameters tk, rh and qv. Distributions of the same parameter may differ with respect to value range or variability.*

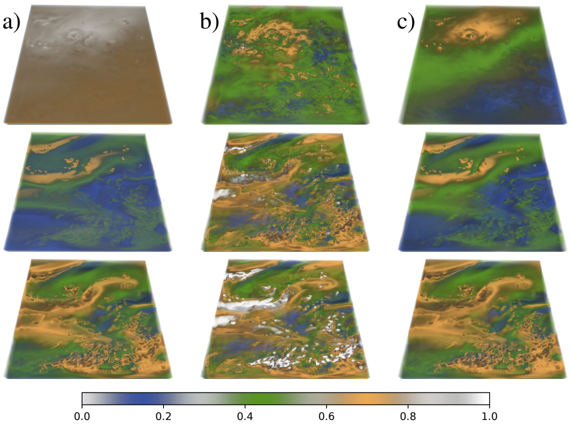

Figure 2: *Influence of the three variants of interval rescaling on the data parameters tk (top row), qv (middle row) and rh (bottom row): a) global min-max, b) local min-max, c) level-wise min-max.*

.

three alternative variants of min-max normalization (see Fig. 2), which reflect a trade-off between expressiveness of the rescaling and storage space required for keeping the meta information:

- Global min-max rescaling: minimum and maximum values are computed over the whole domain, all ensemble members, and all time steps. Minimum and maximum values can be stored as one floating point number each.
- Local min-max rescaling: minimum and maximum values are computed for each grid location separately from the statistics of all ensemble members and all time steps. Minimum and maximum values are stored as a full grid of floating point numbers.
- Level min-max rescaling: minimum and maximum values are computed separately for each height-level in the data. Minimum and maximum values are stored as one-dimensional arrays of floating point values.

4. Model design

(V-)SRNs, in their basic form, are fully-connected neural networks that define a parametric mapping from 3D position coordinates to the d-dimensional data domain [MON*19, CZ19, PFS*19]. To enable sharing of model parameters between different ensemble members, we consider generalized V-SRN mappings, which receive information about the member identity as an additional input.

Encoding of the spatial coordinates For our experiments, we assume that position coordinates are normalized to have values in $[0,1]^3$. The analyzed V-SRN architectures can be subdivided into three modules: a constant input encoding, a low-resolution grid of trainable feature vectors, and a compact fully-connected auto-decoder. For the input encoding, we use Fourier features [MST*20], which map the position coordinates, $\mathbf{p} = (p_x, p_y, p_z) \in [0,1]^3$, to wave-like features

$$\mathbf{f}_{ij} = \left(\sin(2\pi\nu_i\,\mathbf{n}_j\mathbf{p}), \cos(2\pi\nu_i\,\mathbf{n}_j\mathbf{p})\right)$$

with frequency scales $\nu_i = 2^i$, $i \in \mathbb{N}$, and axis-aligned unit directions $\mathbf{n}_j$, $j \in \{x,y,z\}$. Note here, that similar embeddings with randomly chosen frequencies and orientations have been proposed by Tancik *et al.* [TSM*20], but did not yield better results in our experiments. Additionally, we utilize an axis-aligned, regular grid of multi-dimensional feature vectors [CLI*20]. The grid has a pre-set coarse spatial resolution (compared to the resolution of the original data grid) and captures non-local variability in the data. During inference, the features are interpolated trilinearly to match the input position. The network weights and the feature grid are optimized jointly during training. More elaborate multi-resolution feature grids have been proposed recently [MESK22], but were found to not improve the compression-accuracy trade-off of our architectures. For the auto-decoder network, we employ multi-layer perceptrons (MLPs) with l fully-connected layers with c hidden channels. Each layer performs an affine transformation with non-linear activation. Following Weiss *et al.* [WHW21], we use the *SnakeAlt* activation in all but the last layer, and discretize the network weights using half-precision floats and the latent grid using 8 bit per channel. Multi-parameter data is represented by augmenting the output dimension of the decoder models.

Encoding of the ensemble dimension To inform the V-SRN about which ensemble member to reproduce, we explore two different ensemble encoding strategies. First, a separate grid of feature vectors is allocated for each ensemble member and the auto-decoder network is shared between ensemble members. This is similar to how the time dimension is encoded in fV-SRN [WHW21], and replicates the approach of Park *et al.* [PFS*19] in the limit of vanishing spatial resolution. We term this architecture the *multi-grid* configuration. Second, we consider SRNs with a single feature grid, which is shared among all ensemble members, and a separate auto-decoder per ensemble member. The intuition is, that the ensemble information is stored in the shared feature grid, and the separate decoders learn to extract member-specific features from the common grid, thus allowing for efficient reuse of model parameters. We term this variant the *multi-decoder* configuration. As a baseline comparison method, we consider training a separate V-SRN with a single decoder and a single feature grid for every member.

Training At training time, we draw 6×10^6 random positions uniformly distributed in $[0,1]^3$ and sample the original member vol-

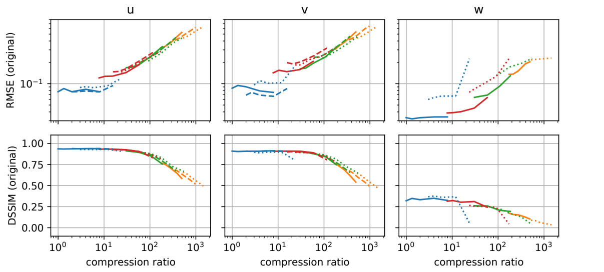

Figure 3: *Reconstruction accuracy of multi-parameter models for parameters u, v and w. Results are shown for single-parameter models (solid lines), two-parameter models (u and v, dashed lines), and three-parameter models (u, v and w, dotted lines). Colors indicate different model configurations, chosen appropriately to have good reconstruction accuracy at each compression ratio.*

umes using trilinear interpolation. We choose a fixed number of samples per ensemble member proportional to the number of samples in the original data grid. The network predictions are matched against the ground truth using the L_1 loss function and stochastic gradient descent. In every mini-batch, we balance the number of samples evenly between all ensemble members to ensure equally distributed gradient variances for all member models. We use the Adam optimizer with an initial learning rate of 10^{-2} and learning rate decay of 0.2 after every 20 epochs. Training lasts for a total of 50 epochs with resampling of the training data applied after every 10 epochs. Loss-adaptive resampling strategies, as described by Weiss *et al.* [WHW21], were found to increase training stability for high-capacity models and slightly improve overall model accuracy. Switching to L_2 loss or omitting the balanced sample distribution among ensemble members led to inferior results.

5. Single-member experiments

To assess the performance of V-SRNs, we first examine the reconstruction accuracy of models which are trained to represent parameter fields from single ensemble members, without accounting for the ensemble dimension. To guarantee proper gradient backpropagation, we fix the decoder architecture as a three-layer MLP and vary the number of channels per layer as well as the resolution and the number of channels in the feature grid. Models were trained separately for dataset parameters tk, rh, qv, u, v and w, and separately for multiple ensemble members. Exemplary compression-accuracy curves for parameters u, v and w with level-wise min-max normalization are shown in Fig. 3 (solid lines).

Impact of model parameterization For all parameters, we find that the details of the decoder architecture have a minor effect on the reconstruction accuracy compared to the parameterization of the latent grid, which is consistent with prior work by Weiss *et al.* [WHW21]. We note that the compression and reconstruction performance of the models depends crucially on an appropriate choice of the grid resolution in horizontal and level direction. Given a fixed grid resolution, we observe a sigmoid-shaped dependence of the reconstruction accuracy on the number of grid feature channels. This indicates that an increasing number of grid channels can partially compensate for a reduction in spatial grid resolution, but not indefinitely. This behavior is observed in qualitatively the same way for various decoder complexities.

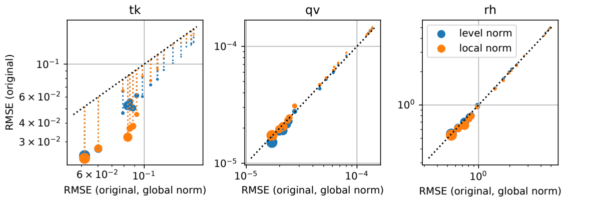

Figure 4: *Impact of data normalization on reconstruction accuracy (RMSE) for models with varying complexity. Point size indicates model complexity (larger point → bigger model). Global min-max normalization is considered as baseline. Points below the dashed diagonal line indicate an improvement.*

Impact of data normalization To evaluate the impact of data normalization on the training outcomes, we retrain single-parameter models on target data to which we apply different normalization schemes. We consider global min-max normalization as a baseline and investigate the effect of applying level-wise or local min-max normalization instead. Specifically, we train model configurations with a three-layer MLP ($c = 32$), and set the latent grid resolution to a fraction of $1/2$ to $1/8$ of the original data grid in all directions. We consider grid feature dimensions of 4 or 8. Fig. 4 illustrates the outcome of such experiments for three parameters with different statistical distributions in height (cf. Fig. 1). For qv and rh, only minimal improvements can be observed from global min-max normalization to level-wise min-max, independent of the model configuration. For tk, which exhibits a much stronger variation of value distribution with height (see Fig. 1, left), both local and level-wise min-max normalization help to reduce the reconstruction error. Local normalization performs better than level-wise min-max only for models with high parameter complexity. We attribute this to the fact that local normalization improves uniformity of the data, but potentially destroys spatial coherence patterns due to high-frequency components in the minimum- and maximum-value fields (see Fig. 2, middle column). Due to the preferable compression rate vs. accuracy trade-off, we use level-wise min-max normalization as a default for all further experiments. More generally, we conjecture that the importance of appropriate data normalization arises due to the inability of the L_1 loss function to properly resolve multi-scale effects. For the field parameter tk, the pronounced field gradient in the vertical direction provides a strong learning signal, while the variability of the data within each level is weighted as relatively less important. Differences in variability, as seen in parameter qv (see Fig. 1, middle), appear less problematic.

Multi-parameter models To evaluate the ability of V-SRNs to fit multi-parameter data, we select a triplet of field parameters – u, v and w, i.e. 3D wind components – for which strong inter-parameter correlations can be expected due to physical reasoning. We train model configurations of different complexity on predict-

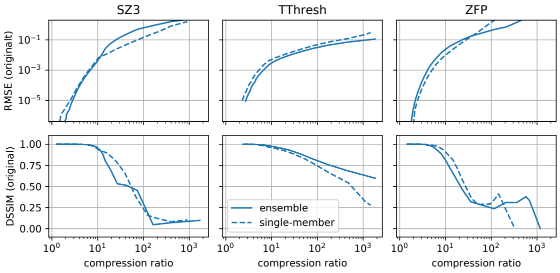

Figure 5: *Reconstruction accuracy vs. compression rate for classical compression algorithms, applied to ensemble data separately for each 3D member volume (solid line) or to a 4D array of stacked member volumes (dashed line).*

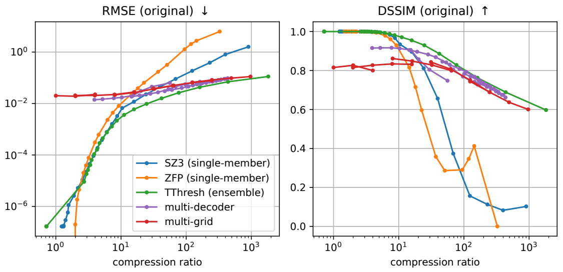

Figure 6: *Comparison of compression algorithms, averaged over all ensemble members. For the baseline compression methods, the best configuration from Fig. 5 are selected. Every point represents a trained network or an invocation of a baseline compression method. Arrows indicate improving quality.*

ing single parameters one at a time, all jointly, or only the horizontal winds. We use model configurations identical to those of the single-parameter experiments, except for adapting the final model layer to the number of required model outputs. Results of the trainings are shown in Fig. 3. The multi-parameter models show qualitatively the same accuracy-compression trade-off as the single parameter models. In particular, the grid parameterization is found to be more important than the decoder complexity. Models with wider decoder layers did not yield higher accuracy than shown in Fig. 3. At low compression ratios (blue curves), u and v are predicted best by the two-parameter model, which suggests that knowledge of both parameters supports accurate reconstruction. At the same time, the three-parameter configuration yields the largest reconstruction error, indicating that joint prediction of unsuitable pairs may hamper high reconstruction accuracy. The parameter is difficult to predict even by single-parameter models, as seen from the low DSSIM values, and thus disturbs the reconstruction of u and v. Only at very high compression ratios, the three-parameter model yields the highest reconstruction accuracy on all parameters.

6. Ensemble experiments

For all subsequent experiments, we use a subset of 64 members of the original ensemble, if not stated otherwise. Experiments are carried out using data for the parameter tk, subject to level-wise min-max normalization.

Classical compression baseline To set a baseline for achievable compression ratios from parameter sharing in the ensemble dimension, we select three commonly used compression algorithms from the literature and evaluate compression performance for ensemble member volumes compressed separately and jointly. We choose SZ3 [DC16] as an example of predictor-based compression algorithms, ZFP [Lin14] as an algorithm with block-wise transform coding, and TThresh [BRLP19], which is based on the Tucker decomposition of tensor data. To allow for a fair comparison between the algorithms, we apply all algorithms with a suitable set of thresholds on absolute error, record the achieved compression ratio and measure the resulting reconstruction accuracy in terms of root mean square error (RMSE) and DSSIM [BPH22]. To evaluate the ability of compression algorithms of exploiting inter-member similarities, we propose a test setting, where an ensemble of 3D volumetric scalar fields is first compressed in a member-by-member configuration (i.e. one 3D volume at a time) and subsequently with all members in common. The comparison of the required storage space per member allows to draw conclusions about whether similarities between ensemble members are exploited efficiently.

Fig. 5 depicts the trade-off between reconstruction accuracy vs. compression ratio found in this procedure. ZFP does not take advantage of between-member similarities. For most accuracy settings, a higher compression ratio is obtained when the 3D volumes are compressed separately. ZFP generally yields poor quality for compression ratios above 30x, but single-member compression is generally preferable. Ensemble compression is favorable with the SZ3 algorithm, in the case of low error thresholds and low-ratio compression. For intermediate and highly lossy compression single-member compression yields lower errors at a given compression ratio. We therefore select the single-member configuration as a baseline for comparison against V-SRN models. TThresh yields overall the best reconstruction accuracy, and is the only algorithm to take advantage from the ensemble dimension throughout the whole range of reconstruction accuracies. We therefore select the ensemble-wise compression for further comparisons.

Ensemble V-SRNs We apply both V-SRN configurations under the same conditions as the classical compressors. For both architectures, we train model variants with three- and four-layer MLP decoders and 32, 64 and 128 channels per layer, and find that models with four layers and 32 channels yield the best balance between reconstruction quality and compression rate. For multi-decoder models, higher decoder capacity is needed to achieve good accuracy, in comparison to V-SRNs trained on single member volumes. This can be seen as a consequence of sharing local feature vectors between multiple decoders. Due to the decoder being unique for every member, increases in decoder size limit the achievable compression ratio. For the multi-grid models, we note that four-layer MLPs with 32 channels yield similar reconstruction accuracy as three-layer architectures with 64 channels, at less than half the storage cost. Further increase of decoder complexity led to only marginal accuracy improvements at significant additional storage cost. Given a fixed

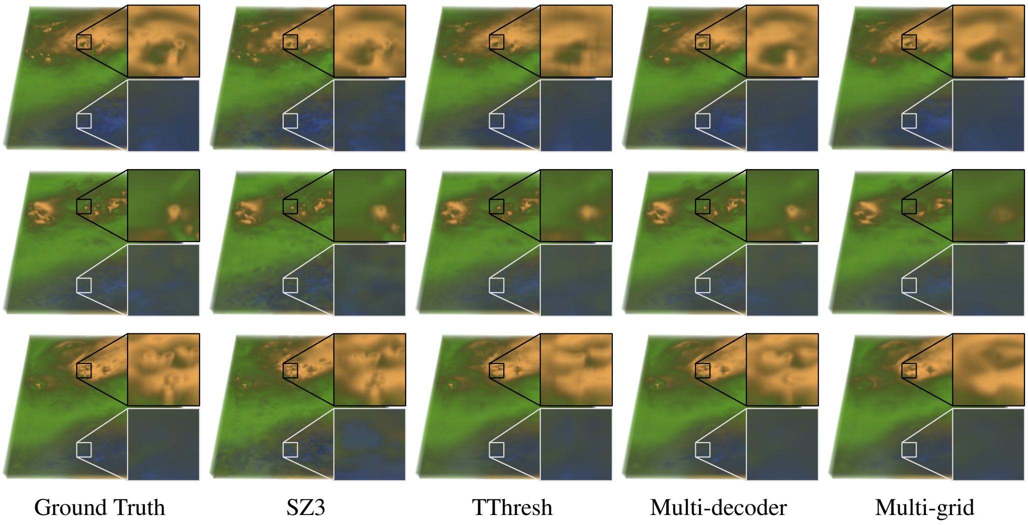

Figure 7: *Qualitative comparison when compressing three member volumes of the tk parameter using SZ3, TThresh, and our two methods "multi-decoder" and "multi-grid". The models were trained on 64 ensemble members, the first three are shown along the rows. Compression ratios of the methods are as follows: SZ3: 248.81x, TThresh: 253.25x, Multi-decoder: 251.88x, Multi-grid: 248.00x. For colorbar, see Fig. 2.*

decoder configuration, feature grid resolution and channel number of the investigated architectures are determined empirically to optimize reconstruction accuracy.

Fig. 6 illustrates the complexity-accuracy trade-off for different configurations of both architectures in comparison to the selected variants (single-member or ensemble-wise compression) of the classical compressors. Multi-grid models allow for higher compression ratios in our test because the majority of parameters is concentrated in the feature grid, which is stored in 8 bit format, thus requiring only half the memory space of an identical number of half-precision network parameters. The accuracy of multi-grid models reaches an optimum around compression ratios of 10x, and is limited by stochastic noise in the optimization of the grid parameters at lower compression ratios. For intermediate and high compression ratios, above 20x, both model variants outperform SZ3 and ZFP in reconstruction accuracy with respect to both RMSE and DSSIM. Given a fixed storage budget, the multi-decoder configuration achieves slightly better reconstruction accuracy. At compression ratios above 200x, both architectures come close to the accuracy of TThresh.

In Fig. 7, we compare the visual quality of renderings of three ensemble members, obtained from reconstructions with compression ratios around 250x. ZFP has been omitted from this comparison due to very low reconstruction quality. The multi-decoder model preserves visual high-frequency structures the best. Both TThresh and the multi-grid model show a tendency to smooth fine-scale details, with TThresh additionally introducing stripe-like artifacts. SZ3 is found to preserve high-frequency field structures in regions of high variability, but introduces fine-granular noise in regions, where the fields should be smooth. The V-SRN models, in contrast rather have a tendency to smooth out fine details, which can be seen as another advantage, depending on the subsequent analysis task. A significant advantage of the V-SRN-based approaches lies in their decompression speed. To reconstruct the full-resolution voxel grid, the reference implementations of TThresh and SZ3 require 50 ms and 10 ms, respectively, on an Ubuntu 20.04 workstation with Intel Xeon W-2133 CPU (3.60 GHz), 32 GB RAM, and Nvidia Titan RTX GPU. Our proposed multi-grid and multi-decoder models sample the full-resolution data in less than 2 ms, and allow for rendering and random data access directly out of the compressed data structure.

We note that the performance benefit of V-SRNs over classical compression algorithms in our application appears comparatively smaller at first sight than was reported in earlier works, such as [WHW21,LJLB21]. We attribute this to the properties of the data that we use for our experiments. The meteorological data differs from previously studied datasets with respect to data size and distribution of variability. In particular the low voxel number and high-frequent variability in the vertical direction prevent the grid-based V-SRNs from achieving higher reconstruction accuracy, because subsampling of the feature grid vertically impedes reconstruction accuracy. Additionally, many of the datasets in earlier studies possess areas of constant field values. Closest to our example is the Hurricane Isabel dataset [isa] as studied by Lu *et al.* [LJLB21], finding that V-SRNs perform similar to TThresh at compression rates around 500x. We expect larger storage savings for simulation data at sub-kilometer resolution, where the fields are determined by low-frequent variability, and for data with higher resolution in the vertical direction, which would simplify the exploitation of data coherence along a third spatial dimension.

Generalization to new ensemble members In Fig. 8, we investigate whether the shared representations of the proposed models encode information that is representative of the full ensemble. For this, we re-used the trained models from previous experiments, fixed the parameterization of the shared model components, and retrained the member-specific components from scratch. Multi-

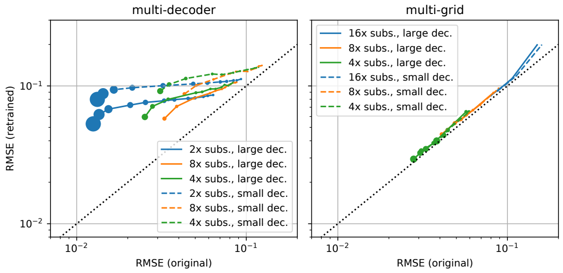

Figure 8: *Reconstruction accuracy on unseen ensemble members after retraining of only the member-specific model parts for different model configurations. Marker size encodes model complexity, black dotted line indicates identity.*

decoder models with higher-capacity decoders (solid lines) achieve better reconstruction accuracy than models with simpler decoders (dashed lines). Nevertheless, all multi-decoder models fail to retain the same accuracy for new members. Models with higher complexity in the latent features perform comparatively worse in fitting unseen ensemble members. The pattern is apparent for model configurations that build on the highest-resolution grid in the test (Fig. 8, blue lines, 2x subsampling), and which exhibit the largest reconstruction error compared to the remaining configurations. We conjecture that a lack of complexity in the latent grid forces the models to learn more abstract and generalizable representations, thus providing better starting conditions for training on unseen members. For the multi-grid configuration, all models are able to achieve almost identical loss levels on unseen members as on the original member set, which confirms the intuition that member-specific information is stored in the feature grids.

Impact of ensemble size Experiments with different numbers of ensemble members were conducted for ensemble sizes between 2 and 128 members. The results indicate that multi-grid models are not affected significantly by changes in ensemble size, suggesting once more that member-specific information is stored in the feature grids. Multi-decoder models yield comparable accuracy for various ensemble sizes at equal compression rates, as well. However, for large ensembles, the evaluation of multi-decoder models is constrained by the memory capacity of the GPU, since the shared feature grid for the full ensemble must be held in device memory, or streamed from system memory or disk.

7. Conclusion

We have analyzed how volume scene representation networks (V-SRNs) can be used to transform a meteorological multi-parameter ensemble into compact neural data representations. We compared two model architectures, which exploit relationships between different field parameters and between ensemble members. Our findings suggest that V-SRNs, in particular in the multi-grid configuration (see Sec. 4), yield promising performance at high compression ratios, where they outperform the classical compressors SZ3 or ZFP in reconstruction accuracy. We found that in meteorological applications the accuracy of V-SRNs may be affected by the choice of hyper parameters and peculiarities of the data distribution. We demonstrate that the latter can be counteracted with appropriate data normalization. However, the necessity of tuning grid resolution and feature channels currently remains a drawback of V-SRN-based data compression. Nevertheless, V-SRNs come with a significant advantage in reconstruction speed and flexibility on multi-parameter data, compared to classical floating-point compressors. This makes them appealing for visual analytics tasks, where an interactive exploration of large multi-parameter ensembles is paramount, using parallelizable statistical evaluations on the whole dataset.

In the future, we intend to shed light on the embedding of network-based compression of multi-parameter ensembles into visual data analysis workflows. For large ensembles comprising billions of data points with many parameters per point, visual analysis techniques like parallel coordinates plots or scatter plot matrices cannot be realized on the GPU due to memory limitations. The fast random access capabilities of V-SRNs allow to overcome these limitations, while at the same time preserving the spatial structure of the data, so that linked 3D spatial data views can be integrated. To improve usability, we will analyze how to design generalizing V-SRNs to limit retraining for new datasets. For this, we consider V-SRNs as a mapping from a latent space representation to an ensemble, and explore speeding up training through direct prediction of the feature representation for new ensembles. Another promising approach could be the combination of V-SRN decoders with generative network architectures, such as variational auto-encoders or generative adversarial networks, which could help to circumvent storage of member-specific feature grids by generating the required features efficiently on demand. Furthermore, we plan on exploring improved methods for hyper-parameter selection, which will enable a higher level of automation and adaptivity, and will improve accessibility of V-SRNs for practical compression applications.

8. Acknowledgments

This study was conducted within the subproject B5 of the Transregional Collaborative Research Center SFB/TRR 165 Waves to Weather funded by the German Research Foundation (DFG).

References

[BHM*16] Baker A. H., Hammerling D. M., Mickelson S. A., Xu H., Stolpe M. B., Naveau P., Sanderson B., Ebert-Uphoff I., Samarasinghe S., De Simone F., et al.: Evaluating lossy data compression on climate simulation data within a large ensemble. *Geoscientific Model Development 9*, 12 (2016), 4381–4403. 1, 2

[BPH22] Baker A. H., Pinard A., Hammerling D. M.: Dssim: a structural similarity index for floating-point data. *arXiv preprint arXiv:2202.02616* (2022). 2, 5

[BRLP19] Ballester-Ripoll R., Lindstrom P., Pajarola R.: Tthresh: Tensor compression for multidimensional visual data. *IEEE transactions on visualization and computer graphics 26*, 9 (2019), 2891–2903. 1, 2, 5

[CDL*19] Cappello F., Di S., Li S., Liang X., Gok A. M., Tao D., Yoon C. H., Wu X.-C., Alexeev Y., Chong F. T.: Use cases of lossy compression for floating-point data in scientific data sets. *The International Journal of High Performance Computing Applications 33*, 6 (2019), 1201–1220. 1

[CLI*20] Chabra R., Lenssen J. E., Ilg E., Schmidt T., Straub J., Lovegrove S., Newcombe R.: Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *European Conference on Computer Vision* (2020), Springer, pp. 608–625. 1, 2, 3

[CZ19] Chen Z., Zhang H.: Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 5939–5948. 1, 2, 3

[DC16] Di S., Cappello F.: Fast error-bounded lossy hpc data compression with sz. In *2016 ieee international parallel and distributed processing symposium (ipdps)* (2016), IEEE, pp. 730–739. 1, 2, 5

[DCG19] Delaunay X., Courtois A., Gouillon F.: Evaluation of lossless and lossy algorithms for the compression of scientific datasets in netcdf-4 or hdf5 files. *Geoscientific Model Development 12*, 9 (2019), 4099–4113. 2

[DLB19] Düben P. D., Leutbecher M., Bauer P.: New methods for data storage of model output from ensemble simulations. *Monthly Weather Review 147*, 2 (2019). 2

[DMG20] Díaz J., Marton F., Gobbetti E.: Interactive spatio-temporal exploration of massive time-varying rectilinear scalar volumes based on a variable bit-rate sparse representation over learned dictionaries. *Computers & Graphics 88* (2020), 45–56. 2

[FM07] Fout N., Ma K.-L.: Transform coding for hardware-accelerated volume rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1600–1607. 2

[GG16] Guthe S., Goesele M.: Variable length coding for gpu-based direct volume rendering. In *Proceedings of the Conference on Vision, Modeling and Visualization* (2016), pp. 77–84. 2

[GIGM12] Gobbetti E., Iglesias Guitián J. A., Marton F.: COVRA: A compression-domain output-sensitive volume rendering architecture based on a sparse representation of voxel blocks. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1315–1324. 2

[HSB*20] Hoang D., Summa B., Bhatia H., Lindstrom P., Klacansky P., Usher W., Bremer P.-T., Pascucci V.: Efficient and flexible hierarchical data layouts for a unified encoding of scalar field precision and resolution. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2020), 603–613. 2

[HW22] Höhlein K., Weiss S.: Evaluation of Volume Representation Networks for Meteorological Ensemble Compression: Code for Experiments, Sept. 2022. doi:10.5281/zenodo.7054427. 2

[HWK*13] Hübbe N., Wegener A., Kunkel J. M., Ling Y., Ludwig T.: Evaluating lossy compression on climate data. In *International Supercomputing Conference* (2013), Springer, pp. 343–356. 2

[isa] Hurricane Isabel dataset, IEEE Visualization challenge 2004. http://sciviscontest-staging.ieeevis.org/2004/data.html. Accessed: 2022-07-05. 6

[KRD*21] Klöwer M., Razinger M., Dominguez J. J., Düben P. D., Palmer T. N.: Compressing atmospheric data into its real information content. *Nature Computational Science 1*, 11 (2021), 713–724. 2

[LCA08] Lee M.-C., Chan R. K., Adjeroh D. A.: Fast three-dimensional discrete cosine transform. *SIAM Journal on Scientific Computing 30*, 6 (2008), 3087–3107. 2

[Lin14] Lindstrom P.: Fixed-rate compressed floating-point arrays. *IEEE transactions on visualization and computer graphics 20*, 12 (2014), 2674–2683. 1, 5

[LJLB21] Lu Y., Jiang K., Levine J. A., Berger M.: Compressive neural representations of volumetric scalar fields. *Computer Graphics Forum 40*, 3 (2021), 135–146. 1, 2, 6

[MAG19] Marton F., Agus M., Gobbetti E.: A framework for gpu-accelerated exploration of massive time-varying rectilinear scalar volumes. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 53–66. 2

[MESK22] Müller T., Evans A., Schied C., Keller A.: Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph. 41*, 4 (July 2022), 102:1–102:15. 2, 3

[MHBB22] Mishra A., Hazarika S., Biswas A., Bryan C.: Filling the void: Deep learning-based reconstruction of sampled spatiotemporal scientific simulation data. doi:10.31219/osf.io/aw7rf. 2

[MLL*21] Martel J. N. P., Lindell D. B., Lin C. Z., Chan E. R., Monteiro M., Wetzstein G.: Acorn: adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics (TOG) 40*, 4 (Aug 2021), 1–13. 2

[MON*19] Mescheder L., Oechsle M., Niemeyer M., Nowozin S., Geiger A.: Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 4460–4470. 1, 2, 3

[MRNK21] Müller T., Rousselle F., Novák J., Keller A.: Real-time neural radiance caching for path tracing. *ACM Trans. Graph. 40*, 4 (Aug. 2021), 36:1–36:16. 2

[MST*20] Mildenhall B., Srinivasan P. P., Tancik M., Barron J. T., Ramamoorthi R., Ng R.: NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision – ECCV 2020* (2020), pp. 405–421. 3

[NGW*20] Necker T., Geiss S., Weissmann M., Ruiz J., Miyoshi T., Lien G.-Y.: A convective-scale 1,000-member ensemble simulation and potential applications. *Quarterly Journal of the Royal Meteorological Society 146*, 728 (2020), 1423–1442. 1, 2

[PFS*19] Park J. J., Florence P., Straub J., Newcombe R., Lovegrove S.: DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 165–174. 1, 2, 3

[SCH*14] Son S. W., Chen Z., Hendrix W., Agrawal A., Liao W.-k., Choudhary A.: Data compression for the exascale computing era. *Supercomputing frontiers and innovations 1*, 2 (2014), 76–88. 1

[SW03] Schneider J., Westermann R.: Compression domain volume rendering. In *IEEE Visualization, 2003. VIS 2003.* (2003), pp. 293–300. 2

[TFT*20] Tewari A., Fried O., Thies J., Sitzmann V., Lombardi S., Sunkavalli K., Martin-Brualla R., Simon T., Saragih J., Niessner M., et al.: State of the art on neural rendering. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 701–727. 2

[TLY*21] Takikawa T., Litalien J., Yin K., Kreis K., Loop C., Nowrouzezahrai D., Jacobson A., McGuire M., Fidler S.: Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 11358–11367. 2

[TSM*20] Tancik M., Srinivasan P., Mildenhall B., Fridovich-Keil S., Raghavan N., Singhal U., Ramamoorthi R., Barron J., Ng R.: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems 33* (2020), 7537–7547. 3

[WHW21] Weiss S., Hermüller P., Westermann R.: Fast neural representations for direct volume rendering. *arXiv preprint* (2021). doi:10.48550/arXiv.2112.01579. 1, 2, 3, 4, 6

[YL95] Yeo B.-L., Liu B.: Volume rendering of dct-based compressed 3d scalar data. *IEEE Transactions on Visualization and Computer Graphics 1*, 1 (1995), 29–43. 2

[ZDL*20] Zhao K., Di S., Liang X., Li S., Tao D., Chen Z., Cappello F.: Significantly improving lossy compression for hpc datasets with second-order prediction and parameter optimization. In *Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing* (New York, NY, USA, 2020), HPDC '20, Association for Computing Machinery, pp. 89–100. 2