The Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-23)

# Learning Similarity Metrics for Volumetric Simulations with Multiscale CNNs

**Georg Kohl, Li-Wei Chen, Nils Thuerey**

Technical University of Munich
{georg.kohl, liwei.chen, nils.thuerey}@tum.de

## Abstract

Simulations that produce three-dimensional data are ubiquitous in science, ranging from fluid flows to plasma physics. We propose a similarity model based on entropy, which allows for the creation of physically meaningful ground truth distances for the similarity assessment of scalar and vectorial data, produced from transport and motion-based simulations. Utilizing two data acquisition methods derived from this model, we create collections of fields from numerical PDE solvers and existing simulation data repositories. Furthermore, a multiscale CNN architecture that computes a volumetric similarity metric (*VolSiM*) is proposed. To the best of our knowledge this is the first learning method inherently designed to address the challenges arising for the similarity assessment of high-dimensional simulation data. Additionally, the tradeoff between a large batch size and an accurate correlation computation for correlation-based loss functions is investigated, and the metric's invariance with respect to rotation and scale operations is analyzed. Finally, the robustness and generalization of *VolSiM* is evaluated on a large range of test data, as well as a particularly challenging turbulence case study, that is close to potential real-world applications.

## 1 Introduction

Making comparisons is a fundamental operation that is essential for any kind of computation. This is especially true for the simulation of physical phenomena, as we are often interested in comparing simulations against other types of models or measurements from a physical system. Mathematically, such comparisons require metric functions that determine scalar distance values as a similarity assessment. A fundamental problem is that traditional comparisons are typically based on simple, element-wise metrics like the $L^1$ or $L^2$ distances, due to their computational simplicity and a lack of alternatives. Such metrics can work reliably for systems with few elements of interest, e.g. if we want to analyze the position of a moving object at different points in time, matching our intuitive understanding of distances. However, more complex physical problems often exhibit large numbers of degrees of freedom, and strong dependencies between elements in their solutions. Those coherences should be considered when comparing physical data, but element-wise operations by definition ignore such interactions between elements. With the curse of dimensionality, this situation becomes significantly worse for systems that are modeled with dense grid data, as the number of interactions grows exponentially with a linearly increasing number elements. Such data representations are widely used, e.g. for medical blood flow simulations (Olufsen et al. 2000), climate and weather predictions (Stocker et al. 2014), and even the famous unsolved problem of turbulence (Holmes et al. 2012). Another downside of element-wise metrics is that each element is weighted equally, which is typically suboptimal; e.g. smoke plumes behave differently along the vertical dimension due to gravity or buoyancy, and small key features like vortices are more indicative of the fluid's general behavior than large areas of near constant flow (Pope 2000).

In the image domain, neural networks have been employed for similarity models that can consider larger structures, typically via training with class labels that provide semantics, or with data that encodes human perception. Similarly, physical systems exhibit spatial and temporal coherence due to the underlying laws of physics that can be utilized. In contrast to previous work on simulation data (Kohl, Um, and Thuerey 2020), we derive an entropy-based similarity model to robustly learn similarity assessments of scalar and vectorial volumetric data. Overall, our work contributes the following:

- We propose a novel similarity model based on the entropy of physical systems. It is employed to synthesize sequences of volumetric physical fields suitable for metric learning.
- We show that our Siamese multiscale feature network results in a stable metric that successfully generalizes to new physical phenomena. To the best of our knowledge this is the first learned metric inherently designed for the similarity assessment of volumetric fields.
- The metric is employed to analyze turbulence in a case study, and its invariance to rotation and scale are evaluated. In addition, we analyze correlation-based loss functions with respect to their tradeoff between batch size and accuracy of correlation computation.

The central application of the proposed *VolSiM* metric is the similarity assessment of new physical simulation methods, numerical or learning-based, against a known ground truth.[1]



[1] Our source code, datasets, and ready-to-use models are available at https://github.com/tum-pbs/VOLSIM. For a version of this work with an appendix also see https://arxiv.org/abs/2202.04109.

This ground truth can take the form of measurements, higher resolution simulations, or existing models. Similar to perceptual losses for computer vision tasks, the trained metric can also be used as a differentiable similarity loss for various physical problems. We refer to Thuerey et al. (2021) for an overview of such problems and different learning methods to approach them.

## 2 Related Work

Apart from simple $L^n$ distances, the two metrics peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) from Wang et al. are commonly used across disciplines for the similarity assessment of data. Similar to the underlying $L^2$ distance, PSNR shares the issues of element-wise metrics (Huynh-Thu and Ghanbari 2008, 2012). SSIM computes a more intricate function, but it was shown to be closely related to PSNR (Horé and Ziou 2010) and thus has similar problems (Nilsson and Akenine-Möller 2020). Furthermore, statistical tools like the Pearson correlation coefficient PCC (Pearson 1920) and Spearman's rank correlation coefficient SRCC (Spearman 1904) can be employed as a simple similarity measurement. There are several learning-based metrics specialized for different domains such as rendered (Andersson et al. 2020) and natural images (Bosse et al. 2016), interior object design (Bell and Bala 2015), audio (Avgoustinakis et al. 2020), and haptic signals (Kumari, Chaudhuri, and Chaudhuri 2019).

Especially for images, similarity measurements have been approached in various ways, but mostly by combining deep embeddings as perceptually more accurate metrics (Prashnani et al. 2018; Talebi and Milanfar 2018). These metrics can be employed for various applications such as image super-resolution (Johnson, Alahi, and Fei-Fei 2016) or generative tasks (Dosovitskiy and Brox 2016). Traditional metric learning for images typically works in one of two ways: Either, the training is directly supervised by learning from manually created labels, e.g. via two-alternative forced choice where humans pick the most similar option to a reference (Zhang et al. 2018), or the training is indirectly semi-supervised through images with class labels and a contrastive loss (Chopra, Hadsell, and LeCun 2005; Hadsell, Chopra, and LeCun 2006). In that case, triplets of reference, same class image, and other class images are sampled, and the corresponding latent space representations are pulled together or pushed apart. We refer to Roth et al. (2020) for an overview of different training strategies for learned image metrics. In addition, we study the behavior of invariance and equivariance to different transformations, which was targeted previously for rotational symmetries (Weiler et al. 2018; Chidester et al. 2019) and improved generalization (Wang, Walters, and Yu 2021).

Similarity metrics for simulation data have not been studied extensively yet. Siamese networks for finding similar fluid descriptors have been applied to smoke flow synthesis, where a highly accurate similarity assessment is not necessary (Chu and Thuerey 2017). Um et al. (2017; 2021) used crowd-sourced user studies for the similarity assessment of liquid simulations which rely on relatively slow and expensive human evaluations. Scalar 2D simulation data was previously compared with a learned metric using a Siamese network (Kohl, Um, and Thuerey 2020), but we overcome methodical weaknesses and improve upon the performance of their work. Their *LSiM* method relies on a basic feature extractor based on common classification CNNs, does not account for the long-term behavior of different systems with respect to entropy via a similarity model during training, and employs a simple heuristic to generate suitable data sequences.

## 3 Modeling Similarity of Simulations

To formulate our methodology for learning similarity metrics that target dissipative physical systems, we turn to the fundamental quantity of entropy. The second law of thermodynamics states that the entropy $S$ of a closed physical system never decreases, thus $\Delta S \geq 0$. In the following, we make the reasonable assumption that the behavior of the system is continuous and non-oscillating, and that $\Delta S > 0$.[2] Eq. 1 is the Boltzmann equation from statistical mechanics (Boltzmann 1866), that describes $S$ in terms of the Boltzmann constant $k_b$ and the number of microstates $W$ of a system.[3]

$$S = k_B \log(W) \tag{1}$$

Since entropy only depends on a single system state, it can be reformulated to take the relative change between two states into account. From an information-theoretical perspective, this is related to using Shannon entropy (Shannon 1948) as a diversity measure, as done by Rényi (1961). Given a sequence of states $s_0, s_1, \ldots, s_n$, we define the relative entropy

$$\tilde{S}(\boldsymbol{s}) = k \log(10^c \boldsymbol{w}_s). \tag{2}$$

Here, $\boldsymbol{w}_s$ is the monotonically increasing, relative number of microstates defined as 0 for $s_0$ and as 1 for $s_n$. $10^c > 0$ is a system-dependent factor that determines how quickly the number of microstates increases, i.e. it represents the speed at which different processes decorrelate. As the properties of similarity metrics dictate that distances are always non-negative and only zero for identical states, the lower bound in Eq. 2 is adjusted to 0, leading to a first similarity model $\hat{D}(\boldsymbol{s}) = k \log(10^c \boldsymbol{w}_s + 1)$. Finally, relative similarities are equivalent up to a multiplicative constant, and thus we can freely choose $k$. Choosing $k = 1/(\log 10^c + 1)$ leads to the full similarity model

$$\mathrm{D}(\boldsymbol{s}) = \frac{\log(10^c \boldsymbol{w}_s + 1)}{\log(10^c + 1)}. \tag{3}$$

For a sequence $\boldsymbol{s}$, it predicts the overall similarity behavior between the reference $s_0$ and the other states with respect to entropy, given the relative number of microstates $\boldsymbol{w}_s$ and the system decorrelation speed $c$.

Fig. 1 illustrates the connection between the logarithmically increasing entropy and the proposed similarity model

[2] These assumptions are required to create sequences with meaningful ground truth distances below in Sec. 4.

[3] We do not have any a priori information about the distribution of the likelihood of each microstate in a general physical system. Thus, the Boltzmann entropy definition which assumes a uniform microstate distribution is used in the following, instead of more generic entropy models such as the Gibbs or Shannon entropy.

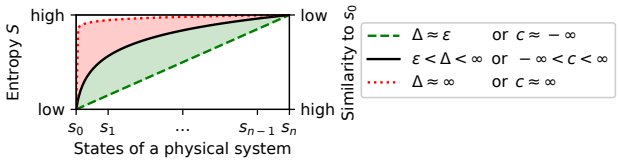


Figure 1: Idealized model of the behavior of entropy and similarity for different $\Delta$ or different $c$, respectively.

for a state sequence with fixed length $n$. Here, $\Delta$ denotes the magnitude of change between the individual sequence states which is directly related to $\boldsymbol{w}_s$, and $c$ is the decorrelation speed of the system that produced the sequence. In the following, we will refer to the property of a sequence being informative with respect to a pairwise similarity analysis as *difficulty*. Sequences that align with the red dotted curve contain little information as they are dissimilar to $s_0$ too quickly, either because the original system decorrelates too fast or because the difference between each state is too large (*high difficulty*). On the other hand, sequences like the green dashed curve are also not ideal as they change too little, and a larger non-practical sequence length would be necessary to cover long-term effects (*low difficulty*). Ideally, a sequence $\boldsymbol{s}$ employed for learning tasks should evenly exhibit both regimes as well as intermediate ones, as indicated by the black curve. The central challenges now become finding sequences with a suitable magnitude of $\Delta$, determining $c$, and assigning distances $\boldsymbol{d}$ to pairs from the sequence.

## 4 Sequence Creation

To create a sequence $s_0, s_1, \ldots, s_n$ within a controlled environment, we make use of the knowledge about the underlying physical processes: We either employ direct changes, based on spatial or temporal coherences to $s_0$, or use changes to the initial conditions of the process that lead to $s_0$. As we can neither directly determine $c$ nor $\boldsymbol{d}$ at this point, we propose to use proxies for them during the sequence generation. Initially, this allows for finding sequences that roughly fall in a suitable difficulty range, and accurate values can be computed afterwards. Here, we use the mean squared error (MSE) as a proxy distance function and the PCC to determine $c$, to iteratively update $\Delta$ to a suitable range.

Given any value of $\Delta$ and a corresponding sequence, pairwise proxy distances[4] between the sequence elements are computed $\boldsymbol{d}^{\Delta} = \mathrm{MSE}(s_i, s_j)$ and min-max normalized to $[0, 1]$. Next, we determine a distance sequence corresponding to the physical changes over the states, which we model as a simple linear increase over the sequence $\boldsymbol{w}_s = (j - i)/n$ following (Kohl, Um, and Thuerey 2020). To indirectly determine $c$, we compare how both distance sequences differ in terms of the PCC as $r = \mathrm{PCC}(\boldsymbol{d}^{\Delta}, \boldsymbol{w}_s)$. We empirically determined that correlations between $0.65$ and $0.85$ work well for all cases we considered. In practice, the network stops learning effectively for lower correlation values as states are too different, while sequences with higher values reduce generalization as a simple metric is sufficient to describe them. Using these thresholds, we propose two semi-automatic iterative methods to create data, depending on the method to introduce variations to a given state (see Fig. 2). Both methods sample a small set of sequences to calibrate $\Delta$ to a suitable magnitude and use that value for the full data set. Compared to strictly sampling every sequence, this method is computationally significantly more efficient as less sampling is needed, and it results in a more natural data distribution.

[4]To keep the notation clear and concise, sequentially indexing the distance vectors $\boldsymbol{d}^{\Delta}$ and $\boldsymbol{w}_s$ with i and j is omitted here.

**[A] Variations from Initial Conditions of Simulations** Given a numerical PDE solver and a set of initial conditions or parameters $\boldsymbol{p}$, the solver computes a solution to the PDE over the time steps $t_0, t_1, \ldots, t_t$. To create a larger number of different sequences, we make the systems non-deterministic by adding noise to a simulation field and randomly generating the initial conditions from a given range. Adjusting *one* of the parameters $p_i$ in steps with a small perturbation $\Delta_i$, allows for the creation of a sequence $s_0, s_1, \ldots, s_n$ with decreasing similarity to the unperturbed simulation output $s_0$. This is repeated for every suitable parameter in $\boldsymbol{p}$, and the corresponding $\Delta$ is updated individually until the targeted MSE correlation range is reached. The global noise strength factor also influences the difficulty and can be updated.

**[B] Variations from Spatio-temporal Coherences** For a source **D** of volumetric spatio-temporal data without access to a solver, we rely on a larger spatial and/or temporal dimension than the one required for a sequence. We start at a random spatio-temporal position $p$ to extract a cubical spatial area $s_0$ around it. $p$ can be repeatedly translated in space and/or time by $\Delta_{t,x,y,z}$ to create a sequence $s_0, s_1, \ldots, s_n$ of decreasing

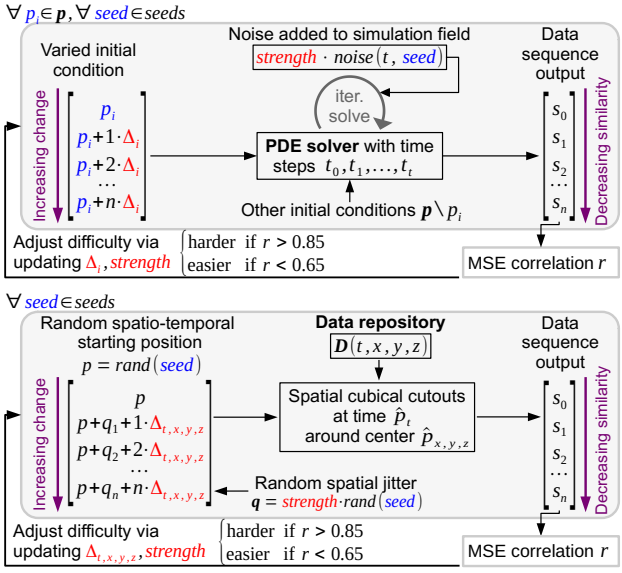


Figure 2: Iteration schemes to calibrate and create data sequences of decreasing similarity. Variation from the reference state can be introduced via the initial conditions of a numerical PDE simulation (method [A], top), or via spatio-temporal data changes on data from a repository (method [B], bottom).

similarity. Note that individual translations in space or time should be preferred if possible. Using different starts leads to new sequences, as long as enough diverse data is available. It is possible to add some global random perturbations $q$ to the positions to further increase the difficulty.

**Data Sets** To create training data with method [A], we utilize solvers for a basic Advection-Diffusion model (`Adv`), Burgers' equation (`Bur`) with an additional viscosity term, and the full Navier-Stokes equations via a Eulerian smoke simulation (`Smo`) and a hybrid Eulerian-Lagrangian liquid simulation (`Liq`). The corresponding validation sets are generated with a separate set of random seeds. Furthermore, we use adjusted versions of the noise integration for two test sets, by adding noise to the density instead of the velocity in the Advection-Diffusion model (`AdvD`) and overlaying background noise in the liquid simulation (`LiqN`).

We create seven test sets via method [B]. Four come from the Johns Hopkins Turbulence Database JHTDB (Perlman et al. 2007) that contains a large amount of direct numerical simulation (DNS) data, where each is based on a subset of the JHTDB and features different characteristics: isotropic turbulence (`Iso`), a channel flow (`Cha`), magneto-hydrodynamic turbulence (`Mhd`), and a transitional boundary layer (`Tra`). Since turbulence contains structures of interest across all length scales, we additionally randomly stride or interpolate the query points for scale factors in $[0.25, 4]$ to create sequences of different physical size. One additional test set (`SF`) via temporal translations is based on ScalarFlow (Eckert, Um, and Thuerey 2019), consisting of 3D reconstructions of real smoke plumes. Furthermore, method [B] is slightly modified for two synthetic test sets: Instead of using a data repository, we procedurally synthesize spatial fields: We employ linearly moving randomized shapes (`Sha`), and randomized damped waves (`Wav`) of the general form $f(x) = cos(x) * e^{-x}$. All data was gathered in sequences with $n = 10$ at resolution $128^3$, and downsampled to $64^3$ for computational efficiency during training and evaluations.

**Determining $c$** For each calibrated sequence, we can now more accurately estimate $c$. As $c$ corresponds to the decorrelation speed of the system, we choose Pearson's distance $d_i^\Delta = 1 - |\mathrm{PCC}(s_0, s_i)|$ as a distance proxy here. $c$ is determine via standard unbounded least-squares optimization from the similarity model in Eq. 3 as $c = \arg\min_c \frac{\log(10^c \, \boldsymbol{d}^\Delta + 1)}{\log(10^c + 1)}$.

## 5 Learning a Distance Function

Given the calibrated sequences $\boldsymbol{s}$ of different physical systems with elements $s_0, s_1, \ldots, s_n$, the corresponding value of $c$, and the pairwise physical target distance sequence $\boldsymbol{w}_s = (j - i)/n$, we can now formulate a semi-supervised learning problem: We train a neural network $m$ that receives pairs from $\boldsymbol{s}$ as an input, and outputs scalar distances $\boldsymbol{d}$ for each pair. These predictions are trained against ground truth distances $\boldsymbol{g} = \frac{\log(10^c \, \boldsymbol{w}_s + 1)}{\log(10^c + 1)}$. Note that $\boldsymbol{g}$ originates from the sequence order determined by our data generation approach, transformed with a non-linear transformation according to the entropy-based similarity model. This technique incorporates the underlying physical behavior by accounting for the decorrelation speed over the sequence, compared to adding variations in a post-process (as commonly done in the domain of images, e.g. by Ponomarenko et al. (2015)). To train the metric network, the correlation loss function in Eq. 4 below compares $\boldsymbol{d}$ to $\boldsymbol{g}$ and provides gradients.

**Network Structure** For our method, we generally follow the established Siamese network structure, that was originally proposed for 2D domains (Zhang et al. 2018): First, two inputs are embedded in a latent space using a CNN as a feature extractor. The Siamese structure means that the weights are shared, which ensures the mathematical requirements for a pseudo-metric (Kohl, Um, and Thuerey 2020). Next, the features from all layers are normalized and compared with an element-wise comparison like an absolute or squared difference. Finally, this difference is aggregated with sum, mean, and learned weighted average functions. To compute the proposed *VolSiM* metric that compares inherently more complex 3D data, changes to this framework are proposed below.

**Multiscale Network** Scale is important for a reliable similarity assessment, since physical systems often exhibit self-similar behavior that does not significantly change across scales, as indicated by the large number of dimensionless quantities in physics. Generally, scaling a data pair should not alter its similarity, and networks can learn such an invariance to scale most effectively by processing data at different scales. One example where this is crucial is the energy cascade in turbulence (Pope 2000), which is also analyzed in our case study below. For learned image metrics, this invariance is also useful (but less crucial), and often introduced with large strides and kernels in the convolutions, e.g. via a feature extractor based on AlexNet (Zhang et al. 2018). In fact, our experiments with similar architectures showed, that models with large strides and kernels generally perform better than models that modify the scale over the course of the network to a lesser extent. However, we propose to directly encode this scale structure in a multiscale architecture for a more accurate similarity assessment, and a network with a smaller resource footprint.

Fig. 3 shows the proposed fully convolutional network: Four scale blocks individually process the input on increasingly smaller scales, where each block follows the same layer structure, but deeper blocks effectively cover a significantly larger volume due to the reduced input resolutions. Deeper

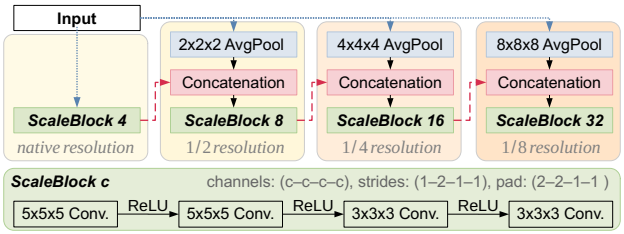


Figure 3: Standard Conv+ReLU blocks (bottom) are interwoven with input and resolution connections (blue dotted and red dashed), to form the combined network architecture (top) with about $350k$ weights.

architectures can model complex functions more easily, so we additionally include resolution connections from each scale block to the next resolution level via concatenation. Effectively, the network learns a mixture of connected deep features and similar representations across scales as a result.

**Training and Evaluation** To increase the model's robustness during training, we used the following data augmentations for each sequence: the data is normalized to $[-1, 1]$, and together randomly flipped and rotated in increments of 90° around a random axis. The velocity channels are randomly swapped to prevent directional biases from some simulations, while scalar data is extended to the three input channels via repetition. For inference, only the normalization operation and the repetition of scalar data is performed. The final metric model was trained with the Adam optimizer with a learning rate of $10^{-4}$ for 30 epochs via early stopping. To determine the accuracy of any metric during inference in the following, we compute the SRCC between the distance predictions of the metric $\boldsymbol{d}$ and the ground truth $\boldsymbol{w}_s$, where a value closer to 1 indicates a better reconstruction.[5]

**Loss Function** Given predicted distances $\boldsymbol{d}$ and a ground truth $\boldsymbol{g}$ of size $n$, we train our metric networks with the loss

$$\begin{aligned} &\mathrm{L}(\boldsymbol{d}, \boldsymbol{g}) = \lambda_1 (\boldsymbol{d} - \boldsymbol{g})^2 + \lambda_2 (1 - r) \\ &\text{where } r = \frac{\sum_{i=1}^{n} (d_i - \bar{d})\,(g_i - \bar{g})}{\sqrt{\sum_{i=1}^{n} (d_i - \bar{d})^2}\,\sqrt{\sum_{i=1}^{n} (g_i - \bar{g})^2}}. \end{aligned} \tag{4}$$

consisting of a weighted combination of an MSE and an inverted correlation term $r$, where $\bar{d}$ and $\bar{g}$ denote the mean. While the formulation follows existing work (Kohl, Um, and Thuerey 2020), it is important to note that $\boldsymbol{g}$ is computed by our similarity model from Sec. 3, and below we introduce a slicing technique to apply this loss formulation to high-dimensional data sets.

To successfully train a neural network, Eq. 4 requires a trade-off: A large batch size $b$ is useful to improve training stability via less random gradients for optimization. Similarly, a sufficiently large value of $n$ is required to keep the correlation values accurate and stable. However, with finite amounts of memory, choosing large values for $n$ and $b$ is not possible in practice. Especially so for 3D cases, where a single sample can already be memory intensive. In general, $n$ is implicitly determined by the length of the created sequences via the number of possible pairs. Thus, we provide an analysis how the correlation can be approximated in multiple steps for a fixed $n$, to allow for increasing $b$ in return. In the following, the batch dimension is not explicitly shown, but all expressions can be extended with a vectorized first dimension. The full distance vectors $\boldsymbol{d}$ and $\boldsymbol{g}$ are split in slices with $v$ elements, where $v$ should be a proper divisor of $n$. For any slice $k$, we can compute a partial correlation $r_k$ with

$$r_k = \frac{\sum_{i=k}^{k+v} (d_i - \bar{d})\,(g_i - \bar{g})}{\sqrt{\sum_{i=k}^{k+v} (d_i - \bar{d})^2}\,\sqrt{\sum_{i=k}^{k+v} (g_i - \bar{g})^2}}. \tag{5}$$

[5]This is equivalent to $\mathrm{SRCC}(\boldsymbol{d}, \boldsymbol{g})$, since the SRCC measures monotonic relationships and is not affect by monotonic transformations, but using $\boldsymbol{w}_s$ is more efficient and has numerical benefits.

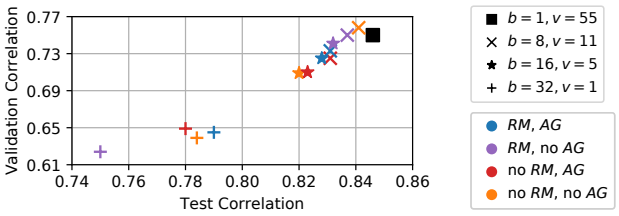


Figure 4: Combined validation and test performance for different batch sizes $b$ and slicing values $v$ (markers), and the usage of running sample mean $RM$ and correlation aggregation $AG$ (colors).

Note that this is only an approximation, and choosing larger values of $v$ for a given $b$ is always beneficial, if sufficient memory is available. For all slices, the gradients are accumulated during backpropagation since other aggregations would required a computational graph of the original, impractical size. Eq. 5 still requires the computation of the means $\bar{d}$ and $\bar{g}$ as a pre-process over all samples. Both can be approximated with the running means $\tilde{d}$ and $\tilde{g}$ for efficiency ($RM$). For small values of $v$, the slicing results in very coarse, unstable correlation results. To alleviate that, it is possible to use a running mean over all previous values $\tilde{r}_k = (1/k)(r_k + \sum_{l=1}^{k-1} r_l)$. This aggregation ($AG$) can stabilize the gradients of individual $r_k$ as they converge to the true correlation value.

Fig. 4 displays the resulting performance on our data, when training with different combinations of $b$, $v$, $RM$, and $AG$. All models exhibit similar levels of memory consumption and were trained with the same random training seed. When comparing models with and without $RM$ both are on par in most cases, even though computation times for a running mean are about 20% lower. Networks with and without $AG$ generalize similarly, however, models with the aggregation exhibit less fluctuations during optimization, leading to an easier training process. Overall, this experiment demonstrates that choosing larger $v$ consistently leads to better results (marker shape), so more accurate correlations are beneficial over a large batch size $b$ in memory-limited scenarios. Thus, we use $b = 1$ and $v = 55$ for the final model.

## 6 Results

We compare the proposed *VolSiM* metric to a variety of existing methods in the upper section of Tab. 1. All metrics were evaluated on the volumetric data from Sec. 4, which contain a wide range of test sets that differ strongly from the training data. *VolSiM* consistently reconstructs the ground truth distances from the entropy-based similarity model more reliably than other approaches on most data sets. As expected, this effect is most apparent on the validation sets since their distribution is closest to the training data. But even on the majority of test sets with a very different distribution, *VolSiM* is the best performing or close to the best performing metric. Metrics without deep learning often fall short, indicating that they were initially designed for different use cases, like *SSIM* (Wang et al. 2004) for images, or variation

| | Validation data sets | | | | Test data sets | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Simulated | | | | Simulated | | Generated | | JHTDB[a] | | | | SF[b] | [c] |
| **Metric** | Adv | Bur | Liq | Smo | AdvD | LiqN | Sha | Wav | Iso | Cha | Mhd | Tra | SF | All |
| *MSE* | 0.61 | 0.70 | 0.51 | 0.68 | 0.77 | 0.76 | 0.75 | 0.65 | 0.76 | **0.86** | 0.80 | 0.79 | 0.79 | 0.70 |
| *PSNR* | 0.61 | 0.68 | 0.52 | 0.68 | 0.78 | 0.76 | 0.75 | 0.65 | **0.78** | **0.86** | **0.81** | 0.83 | 0.79 | 0.73 |
| *SSIM* | **0.75** | 0.68 | 0.49 | 0.64 | 0.81 | 0.80 | 0.76 | 0.88 | 0.49 | 0.55 | 0.62 | 0.60 | 0.44 | 0.61 |
| *VI* | 0.57 | 0.69 | 0.43 | 0.60 | 0.69 | 0.82 | 0.67 | 0.87 | 0.59 | 0.76 | 0.68 | 0.67 | 0.41 | 0.62 |
| *LPIPS (2D)* | 0.63 | 0.62 | 0.35 | 0.56 | 0.76 | 0.62 | 0.87 | 0.92 | 0.71 | 0.83 | 0.79 | 0.76 | 0.87 | 0.76 |
| *LSiM (2D)* | 0.57 | 0.55 | 0.48 | 0.71 | 0.79 | 0.75 | 0.93 | **0.97** | 0.69 | **0.86** | 0.79 | 0.81 | **0.98** | 0.81 |
| *VolSiM (ours)* | **0.75** | **0.73** | **0.66** | **0.77** | **0.84** | **0.88** | **0.95** | 0.96 | 0.77 | **0.86** | **0.81** | **0.88** | 0.95 | **0.85** |
| $CNN_{trained}$ | 0.60 | 0.71 | 0.63 | 0.76 | 0.81 | 0.77 | 0.92 | 0.93 | 0.75 | 0.86 | 0.78 | 0.85 | 0.95 | 0.82 |
| $MS_{rand}$ | 0.57 | 0.66 | 0.45 | 0.69 | 0.76 | 0.75 | 0.80 | 0.78 | 0.74 | 0.86 | 0.80 | 0.82 | 0.84 | 0.74 |
| $CNN_{rand}$ | 0.52 | 0.66 | 0.49 | 0.69 | 0.77 | 0.70 | 0.93 | 0.96 | 0.74 | 0.85 | 0.79 | 0.83 | 0.95 | 0.81 |
| $MS_{identity}$ | 0.75 | 0.71 | 0.68 | 0.73 | 0.83 | 0.85 | 0.87 | 0.96 | 0.74 | 0.87 | 0.77 | 0.87 | 0.94 | 0.82 |
| $MS_{3\ scales}$ | 0.70 | 0.69 | 0.70 | 0.73 | 0.83 | 0.82 | 0.95 | 0.94 | 0.76 | 0.87 | 0.80 | 0.88 | 0.93 | 0.83 |
| $MS_{5\ scales}$ | 0.78 | 0.72 | 0.78 | 0.78 | 0.81 | 0.90 | 0.94 | 0.93 | 0.75 | 0.85 | 0.77 | 0.88 | 0.93 | 0.82 |
| $MS_{added\ \texttt{Iso}}$ | 0.73 | 0.72 | 0.77 | 0.79 | 0.84 | 0.84 | 0.92 | 0.97 | [0.79] | 0.87 | 0.80 | 0.86 | 0.97 | 0.84 |
| $MS_{only\ \texttt{Iso}}$ | 0.58 | 0.62 | 0.32 | 0.63 | 0.78 | 0.65 | 0.72 | 0.92 | [0.82] | 0.77 | 0.86 | 0.79 | 0.65 | 0.75 |

[a] Johns Hopkins Turbulence DB (Perlman et al. 2007) [b] ScalarFlow (Eckert, Um, and Thuerey 2019) [c] Combined test data sets

Table 1: Top: performance comparison of different metrics for 3D data via the SRCC, where values closer to 1 indicate a better reconstruction of the ground truth distances (bold+underlined: best method for each data set, underlined: within a 0.01 margin of the best performing). Bottom: ablation study of the proposed method (brackets: advantage due to different training data).

of information *VI* (Meilă 2007) for clustering. The strictly element-wise metrics *MSE* and *PSNR* exhibit almost identical performance, and both work poorly on a variety of data sets. As the learning-based methods *LPIPS* (Zhang et al. 2018) and *LSiM* (Kohl, Um, and Thuerey 2020) are limited to two dimensions, their assessments in Tab. 1 are obtained by averaging sliced evaluations for all three spatial axes. Both methods show improvements over the element-wise metrics, but are still clearly inferior to the performance of *VolSiM*. This becomes apparent on our aggregated test sets displayed in the `All` column, where *LSiM* results in a correlation value of 0.81, compared to *VolSiM* with 0.85. *LSiM* can only come close to *VolSiM* on less challenging data sets where correlation values are close to 1 and all learned reconstructions are already highly accurate. This improvement is comparable to using *LPIPS* over *PSNR*, and represents a significant step forward in terms of a robust similarity assessment.

The bottom half of Tab. 1 contains an ablation study of the proposed architecture *MS*, and a simple *CNN* model. This model is similar to an extension of the convolution layers of AlexNet (Krizhevsky, Sutskever, and Hinton 2017) to 3D, and does not utilize a multiscale structure. Even though *VolSiM* has more than 80% fewer weights compared to $CNN_{trained}$, it can fit the training data more easily and generalizes better for most data sets in Tab. 1, indicating the strengths of the proposed multiscale architecture. The performance of untrained models $CNN_{rand}$ and $MS_{rand}$ confirm the findings from Zhang et al. (2018), who also report a surprisingly strong performance of random networks. We replace the non-linear transformation of $\boldsymbol{w}_s$ from the similarity model with an identity transformation for $MS_{identity}$ during training, i.e. only the sequence order determines $\boldsymbol{g}$. This consistently lowers the generalization of the metric across data sets, indicating that well calibrated sequences as well as the similarity model are important for the similarity assessment. Removing the last resolution scale block for $MS_{3\ scales}$ overly reduces the capacity of the model, while adding another block for $MS_{5\ scales}$ is not beneficial. In addition, we also investigate two slightly different training setups: for $MS_{added\ \texttt{Iso}}$ we integrate additional sequences created like the `Iso` data in the training, while $MS_{only\ \texttt{Iso}}$ is exclusively trained on such sequences. $MS_{added\ \texttt{Iso}}$ only slightly improves upon the baseline, and even the turbulence-specific $MS_{only\ \texttt{Iso}}$ model does not consistently improve the results on the JHTDB data sets. Both cases indicate a high level of generalization for *VolSiM*, as it was not trained on any turbulence data.

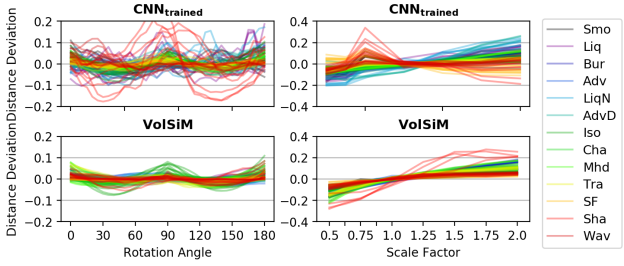


Figure 5: Distance deviation from the mean prediction over differently rotated (left) and scaled (right) inputs for a simple CNN and the proposed multiscale model.

**Transformation Invariance** Physical systems are often characterized by Galilean invariance (McComb 1999), i.e. identical laws of motion across inertial frames. Likewise, a metric should be invariant to transformations of the input, meaning a constant distance output when translating,

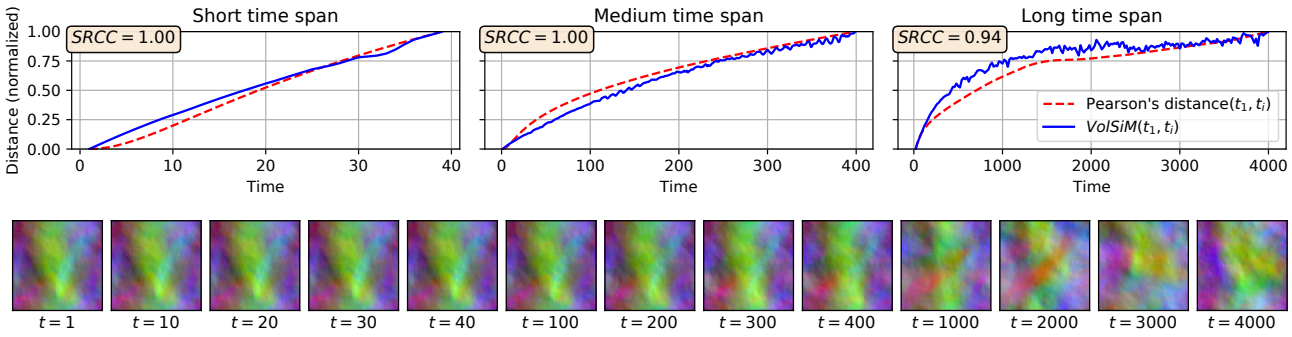


Figure 6: Top: Analysis of forced isotropic turbulence across three time spans. The high SRCC values indicate strong agreement between a traditional correlation evaluation and *VolSiM*. Bottom: Examples from the sequence, visualized via a mean projection along the x-axis and color-coded channels.

scaling, or rotating both inputs. Element-wise metrics fulfill these properties by construction, but our Siamese network structure requires an equivariant feature representation that changes along with input transformations to achieve them. As CNN features are translation equivariant by design (apart from boundary effects and pooling), we empirically examine rotation and scale invariance for our multiscale metric and a standard Conv+ReLU model on a fixed set of 8 random data pairs from each data set. For the rotation experiment, we rotate the pairs in steps of 5° around a random coordinate axis. The empty volume inside the original frame is filled with a value of 0, and data outside the frame is cropped. For scaling, the data is bilinearly up- or downsampled according to the scale factor, and processed fully convolutionally.

In Fig. 5, the resulting distance deviation from the mean of the predicted distances is plotted for rotation and scaling operations. The optimal result would be a perfectly equal distance with zero deviation across all transformations. Compared to the model $CNN_{trained}$, it can be observed that *VolSiM* produces less deviations overall, and leads to significantly smoother and more consistent distance curves, across scales and rotations as shown in Fig. 5. This is caused by the multiscale architecture, which results in a more robust internal feature representation, and thus higher stability across small transformations. Note that we observe scale equivariance rather than invariance for *VolSiM*, i.e. a mostly linear scaling of the distances according to the input size. This is most likely caused by a larger spatial size of the fully convolutional features. Making a scale equivariant model fully invariant would require a form of normalization, which is left for future work.

**Case Study: Turbulence Analysis** As a particularly challenging test for generalization, we further perform a case study on forced isotropic turbulence that resembles a potential real-world scenario for our metric in Fig. 6. For this purpose, fully resolved raw DNS data over a long temporal interval from the isotropic turbulence data set from JHTDB is utilized (see bottom of Fig. 6). The $1024^3$ domain is filtered and reduced to a size of $128^3$ via strides, meaning *VolSiM* is applied in a fully convolutional manner, and has to generalize beyond the training resolution of $64^3$. Three different time spans of the simulation are investigated, where the long span also uses temporal strides. Traditionally, turbulence research makes use of established two-point correlations to study such cases (Pope 2000). Since we are interested in a comprehensive spatial analysis instead of two single points, we can make use of Pearson's distance that corresponds to an aggregated two-point correlation on the full fields to obtain a physical reference evaluation in this scenario.

Fig. 6 displays normalized distance values between the first simulation time step $t_1$ and each following step $t_i$. Even though there are smaller fluctuations, the proposed *VolSiM* metric (blue) behaves very similar to the physical reference of aggregated two-point correlations (red dashed) across all time spans. This is further emphasized by the high SRCC values between both sets of trajectories, even for the challenging long time span. Our metric faithfully recovers the correlation-based reference, despite not having seen any turbulence data at training time. Overall, this experiment shows that the similarity model integrates physical concepts into the comparisons of *VolSiM*, and indicates the generalization capabilities of the multiscale metric to new cases.

## 7 Conclusion

We presented the multiscale CNN architecture *VolSiM*, and demonstrated its capabilities as a similarity metric for volumetric simulation data. A similarity model based on the behavior of entropy in physical systems was proposed and utilized to learn a robust, physical similarity assessment. Different methods to compute correlations inside a loss function were analyzed, and the invariance to scale and rotation transformations investigated. Furthermore, we showed clear improvements upon elementwise metrics as well as existing learned approaches like *LPIPS* and *LSiM* in terms of an accurate similarity assessment across our data sets.

The proposed metric potentially has an impact on a broad range of disciplines where volumetric simulation data arises. An interesting area for future work is designing a metric specifically for turbulence simulations, first steps towards which were taken with our case study. Additionally, investigating learning-based methods with features that are by construction equivariant to rotation and scaling may lead to further improvements in the future.

## Ethical Statement

Since we target the fundamental problem of the similarity assessment of numerical simulations, we do not see any direct negative ethical implications of our work. However, there could be indirect negative effects since this work can act as a tool for more accurate and/or robust numerical simulations in the future, for which a military relevance exists. A further indirect issue could be explainability, e.g. when simulations in an engineering process yield unexpected inaccuracies.

## Acknowledgments

This work was supported by the ERC Consolidator Grant *SpaTe* (CoG-2019-863850).

## References

Andersson, P.; Nilsson, J.; Akenine-Möller, T.; Oskarsson, M.; Åström, K.; and Fairchild, M. D. 2020. FLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 3(2): 15:1–15:23.

Avgoustinakis, P.; Kordopatis-Zilos, G.; Papadopoulos, S.; Symeonidis, A. L.; and Kompatsiaris, I. 2020. Audio-Based Near-Duplicate Video Retrieval with Audio Similarity Learning. In *25th International Conference on Pattern Recognition (ICPR 2020)*, 5828–5835.

Bell, S.; and Bala, K. 2015. Learning Visual Similarity for Product Design with Convolutional Neural Networks. *ACM Transactions on Graphics*, 34(4): 98:1–98:10.

Boltzmann, L. 1866. *Über Die Mechanische Bedeutung Des Zweiten Hauptsatzes Der Wärmetheorie: (Vorgelegt in Der Sitzung Am 8. Februar 1866)*. Staatsdruckerei.

Bosse, S.; Maniry, D.; Mueller, K.-R.; Wiegand, T.; and Samek, W. 2016. Neural Network-Based Full-Reference Image Quality Assessment. In *2016 Picture Coding Symposium (PCS)*.

Chidester, B.; Zhou, T.; Do, M. N.; and Ma, J. 2019. Rotation Equivariant and Invariant Neural Networks for Microscopy Image Analysis. *Bioinformatics*, 35(14): i530–i537.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 539–546. IEEE Computer Society.

Chu, M.; and Thuerey, N. 2017. Data-Driven Synthesis of Smoke Flows with CNN-Based Feature Descriptors. *ACM Transactions on Graphics*, 36(4): 69:1–69:14.

Dosovitskiy, A.; and Brox, T. 2016. Generating Images with Perceptual Similarity Metrics Based on Deep Networks. In *Advances in Neural Information Processing Systems 29*, volume 29.

Eckert, M.-L.; Um, K.; and Thuerey, N. 2019. ScalarFlow: A Large-Scale Volumetric Data Set of Real-World Scalar Transport Flows for Computer Animation and Machine Learning. *ACM Transactions on Graphics*, 38(6).

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 1735–1742.

Holmes, P.; Lumley, J. L.; Berkooz, G.; and Rowley, C. W. 2012. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press. ISBN 978-0-511-91970-1.

Horé, A.; and Ziou, D. 2010. Image Quality Metrics: PSNR vs. SSIM. In *20th International Conference on Pattern Recognition (ICPR 2010)*, 2366–2369.

Huynh-Thu, Q.; and Ghanbari, M. 2008. Scope of Validity of PSNR in Image/Video Quality Assessment. *Electronics Letters*, 44(13): 800–801.

Huynh-Thu, Q.; and Ghanbari, M. 2012. The Accuracy of PSNR in Predicting Video Quality for Different Video Scenes and Frame Rates. *Telecommunication Systems*, 49(1): 35–48.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Computer Vision - ECCV 2016*, volume 9906, 694–711.

Kohl, G.; Um, K.; and Thuerey, N. 2020. Learning Similarity Metrics for Numerical Simulations. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119, 5349–5360.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, 60(6): 84–90.

Kumari, P.; Chaudhuri, S.; and Chaudhuri, S. 2019. PerceptNet: Learning Perceptual Similarity of Haptic Textures in Presence of Unorderable Triplets. In *2019 IEEE World Haptics Conference (WHC)*, 163–168. IEEE.

McComb, W. D. 1999. *Dynamics and Relativity*. Oxford University Press. ISBN 0-19-850112-9.

Meilă, M. 2007. Comparing Clusterings—an Information Based Distance. *Journal of Multivariate Analysis*, 98(5): 873–895.

Nilsson, J.; and Akenine-Möller, T. 2020. Understanding SSIM. *arXiv:2006.13846 [cs, eess]*.

Olufsen, M. S.; Peskin, C. S.; Kim, W. Y.; Pedersen, E. M.; Nadim, A.; and Larsen, J. 2000. Numerical Simulation and Experimental Validation of Blood Flow in Arteries with Structured-Tree Outflow Conditions. *Annals of Biomedical Engineering*, 28(11): 1281–1299.

Pearson, K. 1920. Notes on the History of Correlation. *Biometrika*, 13(1): 25–45.

Perlman, E.; Burns, R.; Li, Y.; and Meneveau, C. 2007. Data Exploration of Turbulence Simulations Using a Database Cluster. In *Proceedings of the ACM/IEEE Conference on High Performance Networking and Computing*, 1–11.

Ponomarenko, N.; Jin, L.; Ieremeiev, O.; Lukin, V.; Egiazarian, K.; Astola, J.; Vozel, B.; Chehdi, K.; Carli, M.; Battisti, F.; and Kuo, C. C. J. 2015. Image Database TID2013: Peculiarities, Results and Perspectives. *Signal Processing-Image Communication*, 30: 57–77.

Pope, S. 2000. *Turbulent Flows*. Cambridge University Press. ISBN 978-0-511-84053-1.

Prashnani, E.; Cai, H.; Mostofi, Y.; and Sen, P. 2018. PieAPP: Perceptual Image-Error Assessment through Pairwise Preference. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1808–1817. IEEE Computer Society.

Rényi, A. 1961. On Measures of Entropy and Information. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, 547–561. University of California Press.

Roth, K.; Milbich, T.; Sinha, S.; Gupta, P.; Ommer, B.; and Cohen, J. P. 2020. Revisiting Training Strategies and Generalization Performance in Deep Metric Learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119, 8242–8252. PMLR.

Shannon, C. E. 1948. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3): 379–423.

Spearman, C. 1904. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology*, 15(1): 72–101.

Stocker, T.; Qin, D.; Plattner, G.-K.; Tignor, M.; Allen, S.; Borschung, J.; Nauels, A.; Xia, Y.; Bex, V.; and Midgley, P. 2014. *Climate Change 2013: The Physical Science Basis*. Cambridge University Press. ISBN 978-1-107-41532-4.

Talebi, H.; and Milanfar, P. 2018. Learned Perceptual Image Enhancement. In *2018 IEEE International Conference on Computational Photography (ICCP)*.

Thuerey, N.; Holl, P.; Mueller, M.; Schnell, P.; Trost, F.; and Um, K. 2021. Physics-Based Deep Learning. https://physicsbaseddeeplearning.org. Accessed: 2023-03-27.

Um, K.; Hu, X.; and Thuerey, N. 2017. Perceptual Evaluation of Liquid Simulation Methods. *ACM Transactions on Graphics*, 36(4).

Um, K.; Hu, X.; Wang, B.; and Thuerey, N. 2021. Spot the Difference: Accuracy of Numerical Simulations via the Human Visual System. *ACM Transactions on Applied Perception*, 18(2): 6:1–6:15.

Wang, R.; Walters, R.; and Yu, R. 2021. Incorporating Symmetry into Deep Dynamics Models for Improved Generalization. In *9th International Conference on Learning Representations (ICLR 2021)*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.

Weiler, M.; Geiger, M.; Welling, M.; Boomsma, W.; and Cohen, T. 2018. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Advances in Neural Information Processing Systems 31*, 10402–10413.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 586–595.