Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

The effects of non-linearity operators in voxel-based 3D neural networks for shape reconstruction

Timo Friedrich, Patricia Wollstadt, Stefan Menzel

2020

Preprint:

This is an accepted article published in IEEE Symposium Series on Computational Intelligence (SSCI). The final authenticated version is available online at: https://doi.org/10.1109/SSCI47803.2020.9308308 Copyright 2020 IEEE

The Effects Of Non-linear Operators In Voxel-Based Deep Neural Networks For 3D Style Reconstruction

Timo Friedrich, Patricia Wollstadt, and Stefan Menzel
Honda Research Institute Europe
Offenbach/Main, Germany
{timo.friedrich, patricia.wollstadt, stefan.menzel}@honda-ri.de

***Abstract*— Neural Style Transfer has been successfully applied for generating plausible novel 2D images by transferring the style of a painting to an existing content image and has been extended to image, video, motion and animation manipulation. Besides first works on 2D/3D hybrid approaches relying on 2D rendering operations, a consistent end-to-end 3D Neural Style Transfer is missing. For realizing such an end-to-end 3D Neural Style Transfer, we suggested a first 3D voxel-based neural network architecture in previous works. Here, we extend this architecture to a resolution of 256^3 voxels to process vastly finer visual features, and we include our recently introduced standardized gram matrix approach, which reduces noise and visually improves style transfer results. We furthermore show that the type of activation function used in the network is crucial for the successful application in 3D Neural Style transfer. In particular, we demonstrate that the recently proposed bipolar exponential linear unit (belu) operator achieves a consistent style reconstruction independent of filter orientation relative to the shape surface. We compare the operator to the commonly used exponential linear units (elu) and provide insights for the shown behavior. In addition to a purely visual inspection, we evaluate the constructed shapes using local 3D shape descriptors and confirm the subjective visual improvements. We conclude that the utilization of belu operators in the deep neural network architecture is a crucial component for realizing high-resolution voxel-based 3D Neural Style Transfer.**

Keywords—Deep neural networks, non-linear operators, neural style transfer, voxel representations

I. Introduction

Neural Style Transfer for generating artificial 2D images [1] is an exciting use case for deep learning methods. We strive for an extension of these methods to the 3D domain for a potential application as part of a man-machine cooperative design system which is capable of applying novel style ideas in an automated fashion to a set of 3D geometries, e.g. cars, and provides valuable feedback to the designer on the appeal of product line-ups for an improved decision-making process.

2D Neural Style Transfer has matured over the last years and been implemented in several software products for image and video manipulation. The original concept of transferring the style of an image onto a content target, e.g., transferring the style of a painting onto a photograph [1], has vastly matured [2], [3], [4] and has even been extended to further domains such as motion data [5]. In our research, we focus on the development of future methods that enable the utilization of style transfer techniques similar to 2D Neural Style Transfer in the 3D domain [6]. Such methods may allow for applications, e.g., in the automotive domain, where they enable a designer or engineer to explore potential solution spaces in the vehicle development process by selecting styles and applying them to 3D shape content for rapid visual assessment and further inspiration. The objective of the current research is the investigation of fundamental principles and methods considering today's computational constraints as a basis for a future, interactive and cooperative stylization pipeline.

3D shape data may be encoded in a variety of different formats (surface polygon meshes, point clouds, voxels), that each have several impacts on 3D Neural Style Transfer [6]. Even though voxels are the most promising representation due to their similarity to pixels, fundamental differences between the 2D and 3D domain remain, which require setup and configuration changes of neural network architectures and style transfer algorithms. We have shown in [7] for 64^3 voxels shapes (Fig. 1, middle) that voxels demand an updated style loss term as well as the choice of appropriate neural network operators. In addition, we concluded that the resolution of 64^3 voxels was insufficient and occludes stylistic features [6], however, an extension to higher resolutions poses significant computational challenges.

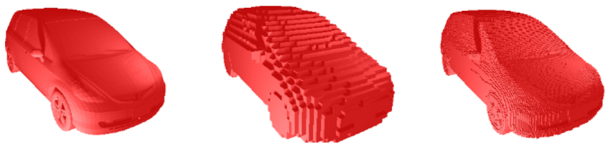


Fig. 1. Left: Triangular mesh. Middle: 64^3 voxel model. Right: 256^3 voxel model. Model taken from ModelNet40 [8]

In the present paper, we extend the previous architecture to 256^3 voxels (Fig. 1, right) and handle computational limitations by carefully selecting appropriate network operators. In particular, we investigate and show the influence of different activation functions for deep classification networks used in the context of 3D Neural Style Transfer. Based on our results, we propose to utilize the recently published bipolar activation function approach [9] in the deep neural network architecture to

978-1-7281-2547-3/20/$31.00 ©2020 IEEE

build the prerequisites for a meaningful end-to-end voxel-based Neural Style Transfer of 3D shapes.

Because of the complexity of the overall system, we validate the effects of different activation functions used in the classification network by performing style reconstruction, which is a sub-optimization within the Neural Style Transfer pipeline, where the content is ignored and shapes are created solely based on artistic style statistics. The resulting 3D models consist of randomly distributed distinct style features associated with a style target and allow for an early assessment of the style encoding capabilities of the used Neural Style Transfer method. In addition to the visual (and eventually subjective) evaluation of the resulting shapes, we used a set of 3D shape descriptors and demonstrate their application for the quantification of style reconstruction success underlining the improved function of our deep neural networks architecture.

The remainder of the paper is organized as follows. In Section II, we provide details on relevant literature for Neural Style Transfer and elaborate the specifics of 3D style transfer, which utilizes our standardized gram matrix approach [7]. Section III outlines our motivation and introduces the derived architectural choices for our voxel-based classification networks. Also, we introduce a quantitative measure to evaluate voxel shapes on style similarity. Section IV builds on top of the previous sections. First, we discuss the performance of our classification networks. Then, we take these models and apply the updated style reconstruction algorithm. We show and solve issues regarding the activation functions used in neural networks for Neural Style Transfer. Furthermore, we evaluate our style reconstructions visually and quantitatively with voxel-based shape descriptors. Section V concludes the paper.

II. State of the Art

We first provide some background on 2D Neural Style Transfer to motivate the use of style reconstruction to evaluate the 3D network architecture presented in the present paper. We then go on to review prior work on 3D style transfer.

A. 2D Neural Style Transfer And Style Reconstruction

Originally, 2D Neural Style Transfer was developed for the transfer of artistic style features of a painting onto another image, e.g., a photograph [1], [2], [3]. Gatys et al. [1] proposed a framework based on the exploitation of deep neural networks. Taking intermediate activation maps of the state-of-the-art VGG19 classification network [10], they suggested an optimization, which iteratively alters a generic image until its content matches the content target image, and the statistic of the style features matches the statistic of a style target image. Both optimize for a total loss $\mathcal{L}_{total}$, which is expressed as a weighted sum of the independent losses for content $\mathcal{L}_{content}$ and style $\mathcal{L}_{style}$:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{content} + \beta\mathcal{L}_{style}. \tag{1}$$

During the style transfer optimization, the value of each pixel x_{ij} of the to be generated image $\vec{x}$ is updated based on the total loss gradient (2), where η denotes a global learning rate.

$$\Delta x_{ij} = -\eta \frac{\partial \mathcal{L}_{total}}{\partial x_{ij}} \tag{2}$$

The content loss, $\mathcal{L}_{content}$, takes the convolutional activation maps F_{ij}^{l} of layer l of the neural network for both, the to be generated image $\vec{x}$ and the content target image $\vec{p}$, to calculate the mean squared error as

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l\,) = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l}(\vec{p}) - F_{ij}^{l}(\vec{x})\right)^2. \tag{3}$$

The style loss, $\mathcal{L}_{style}$, transfers a feature statistic with detached spatial relation and is defined as a weighted sum of the per-layer loss values,

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} \omega_l E_l, \tag{4}$$

where, $\vec{a}$ is the style target and E_l denotes the style loss per chosen style encoding layer, which is in turn calculated as the normalized mean squared error,

$$\mathrm{E}_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}\left(G_{ij}^{l}(\vec{a}) - G_{ij}^{l}(\vec{x})\right)^2, \tag{5}$$

where G_{ij}^{l} are the Gram matrices defined as

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}. \tag{6}$$

In sum, the style loss (4) is made independent of the spatial location by basing it on the gram matrices G_{ij}^{l} of the chosen activation maps. Consequently, the location of artistic style features becomes irrelevant and only their occurrence statistic is important.

The style loss can be optimized independently from the content loss by setting $\alpha = 0$. This produces images without recognizable content but a visual artistic style [1]. In the present paper, we present steps towards a consistent end-to-end 3D Neural Style Transfer and focus first solely on the style reconstruction of 3D objects, which, in comparison to the content reconstruction, is the more challenging optimization part and allows to evaluate neural network architectural changes without the need to realize the complete 3D voxel-based Neural Style Transfer pipeline.

B. Towards 3D Neural Style Transfer

In the following, we present recent research, which relates to the general concept of 3D (non-neural network based) style transfer and hybrid 2D/3D approaches before introducing our proposed end-to-end voxel-based 3D Neural Style Transfer.

A method for non-neural-network based style transfer was

proposed by Ma et al. [11], who introduced a shape analogy driven algorithm. This method, inspired by cognitive science, creates 3D models for geometric shapes like furniture and buildings, but is limited to shapes with clearly separable, repeating sub-structures. Furthermore, we proposed an Eigendecomposition-based style transfer realized for vehicle line-ups [12], which however requires 3D meshes with homogeneous topology (meshes with identical mesh graph structures), such that the method is bound to an a priori homogenized, triangular mesh-based dataset. Thus, while both methods are entirely 3D methods, they are not applicable to the general case of two arbitrary shapes with varying topologies.

With respect to neural network approaches, Kato et al. [13] have recently introduced a hybrid 2D/3D algorithm, which alters the texture and shape of three-dimensional triangular meshes. In the field of computer graphics, Kim et al. [14] proposed a method to transfer styles of images on fluid animations, while Sýkora et al. [15] suggested a technique to transfer style onto 3D animations with proper lighting and good temporal stability. All methods have in common that they rely on the 2D Neural Style Transfer foundations in combination with differentiable image rendering. Hence, the whole process can be optimized using algorithms like gradient descent, so that not only the color and texture of the 3D objects can be altered, but also the vertex positions according to a backpropagated loss signal [13]. However, while these algorithms generate visually appealing results, they lack the ability to process inner occluded structures as it would be required for engineering and design data in the automotive domain. Another 2D/3D approach is proposed by Ren et al. who applied Neural Style Transfer on voxel-based architectural models by a posteriori stacking 2D images [16].

To overcome the gaps of 2D/3D hybrid Neural Style Transfer and non-neural style transfer approaches, we proposed a framework for an entire 3D Neural Style Transfer for voxel representations [6]. We highlighted current challenges, such as low resolutions of 3D inputs, as well as the lack of deep 3D classification networks that are pre-trained on rich and dense datasets as confirmed by Qi et al. [17] and Kim et al. [14].

Furthermore, we showed that down-sampling by convolutional striding greater than two has to be avoided and, instead, max pooling should be used in order to circumvent undesired interactions of the convolutional filters on the voxel shape border areas [7]. Additionally, we showed that our adapted style metric,

$$\mathrm{E}_l = \sum_{i,j} \left(G^l_{ij,s} - A^l_{ij,s}\right)^2, \tag{7}$$

which is based on a standardized gram matrix,

$$G^l_{ij,s} = \frac{G^l_{ij} - \mu\left(G^l_{ij}\right)}{\sigma\left(G^l_{ij}\right)}, \tag{8}$$

generates higher quality results in terms of artifacts, noise and subjective visual feature recognition [7]. Here, μ and σ denote the mean and standard variation of a given tensor. Our results presented in this paper are based on the standardized gram matrix approach.

In addition to all technical differences between 2D image and 3D voxel style transfer, the data itself shows fundamentally different elements of art [18]. E.g., voxel data is usually monochrome or binary, thus lacking color information. Further, texture and composition are hardly applicable to the ModelNet40 dataset [8], which we use to train our classification networks. However, in previous work, we have shown that the original Neural Style Transfer algorithm by Gatys et al. [1] still produces reasonable results for features typically found in voxel models. We applied Gatys' algorithm to 2D shape primitives with art properties similar to those found in voxel models (Fig. 2). Our results indicated the general applicability of Neural Style Transfer to voxel data.

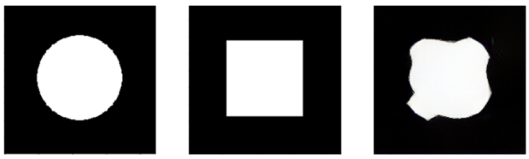


Fig. 2. Style Transfer proof of concept for 2D binary geometric primitives. Left: Style target. Middle: Content target. Right: Style transfer result [7]

III. Methodology

A. Network Architectures

For our overall target of developing an end-to-end 3D Neural Style Transfer pipeline for pure voxel data, we build upon high-quality classification networks. We extended our set of classification networks used in [7] to an input-resolution of 256^3 voxels and applied it to the ModelNet40 dataset, which was voxelized with a resolution of 256^3 voxels using the binvox package [19], [20]. The architectural decisions for these networks were not solely driven by the desire to achieve the best possible classification score, but also by the expected effects on the subsequent Neural Style Transfer.

The increased input size of 256^3 voxel models and the resulting increase in size of corresponding activation maps within the network lead to a high computational demand during training that limits network design choices. In particular, due to memory constraints on the graphics cards, we set the number of filters used by each convolutional layer to 24. This number is

TABLE I
3D Voxel Classification Networks

	Network Variants		
	ELU	***TANH***	***BELU***
FC[1] (Out)	(40) softmax, class labels		
FC	(64) elu		
FC	(128) elu		
Conv[2]_5	(4, 4, 4, 128)		(4, 4, 4, 128*2)
Conv_4	(12, 12, 12, 96)		(12, 12, 12, 96*2)
Conv_3	(28, 28, 28, 64)		(28, 28, 28, 64*2)
Conv_2	(60, 60, 60, 64)		(60, 60, 60, 64*2)
Conv_1	(125, 125, 125, 32)		(125, 125, 125, 32*2)
Conv_0	(254 ,254, 254, 24)		(254 ,254, 254, 12*2)
Input	(256 ,256 ,256 ,1) voxel model		

[1] Fully connected layer
[2] Convolutional layer

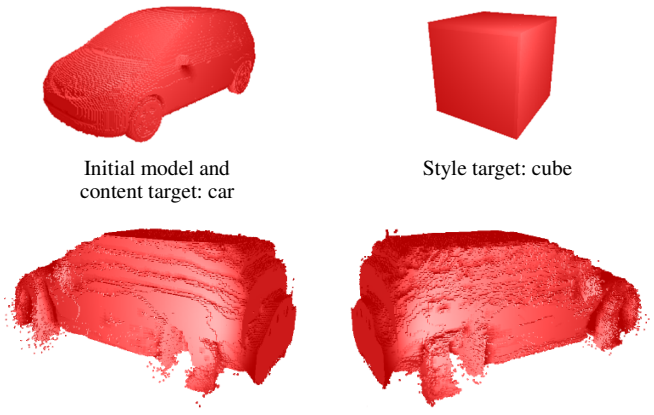
smaller than the number obtained by extrapolating the number of filters used in the VGG19 [10] model to the 3D domain. VGG19 uses 64 filters in the first layers, which would result in 96 filters when accounting for the additional spatial dimension of voxel models. Because of this memory restriction we have to ensure that the filter responses and their further processing by activation functions provide a high quality.

Initially, we started our experiments by keeping the exponential linear unit (elu) [21] as activation function (Table I, left column) in the classification networks and trained them on the aligned ModelNet40 dataset. Early experiments showed that increasing the number of convolutional layers did not yield better classification accuracy. Instead, we obtained the best performance values by keeping the general architecture and adapting the layer parameters in order to use all available graphics card memory (Nvidia RTX 8000, 48 GB) resources for our setup. However, we had to introduce more maxpool operators and to reduce the batch size to comply with computational limitations. Every convolutional layer uses down-sampling with maxpooling with stride 2. Each layer, convolutional and fully connected, applies batch normalization [22].

B. *Effects of Activation Functions on Neural Style Transfer*

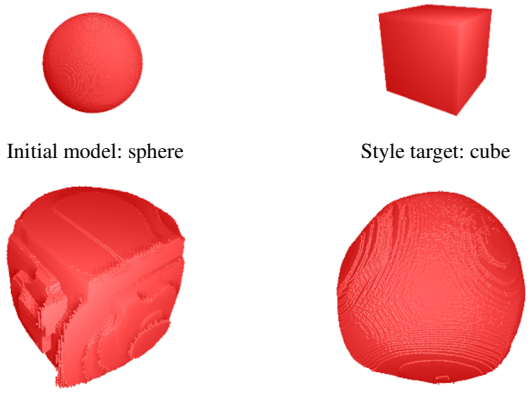
3D voxel-based Neural Style Transfer results for a 3D car model generated with the elu network (Table I, left column) showed different shape features dependent on the view angle (Fig. 3). In the left view, stair-like features were visible, which occurrence was expected since we specified a cube as style target. When viewing the same result from the opposing right side, we noticed the lack of these cubic features and predominantly saw a noisy shape surface. We found similar discrepancies for top/bottom and front/rear views, not only in a single style transfer result, but over a series of proof-of-concept experiments. Over all experiments, the directional dependency remained constant, i.e., always one side showed proper features while the other did not (Fig. 3).

The same "asymmetric" behavior was observable in 3D style reconstructions based on the same elu network setup (Fig. 4). Here, the left view confirms a properly configured



3D Neural Style Transfer result from two different angles

Fig. 3. Voxel-based 3D Neural Style Transfer result based on the elu network. The result shows different shape features on two opposing sides, namely stair like cubic features from the left and noisy random features from the right side. This implied a directional dependency during the Neural Style Transfer optimization.



Style reconstruction result from two different angles

Fig. 4. 3D style reconstruction result depicted from two opposing angles confirming the directional dependency of the elu network based Neural Style Transfer setup. The first view shows cubic features like flat faces and 90° angles, while the second view shows only round features remaining from the initial model.

Neural Style Transfer setup by producing cubic features such as flat faces and 90° edges, while the right view showed none. Since the effect was more pronounced on style reconstructions (Fig. 4) compared to style transfer results (Fig. 3), we based the follow-up investigation and evaluation on the former. Nevertheless, we assume results to be directly applicable to style transfer because style reconstruction is an essential sub-task of Neural Style Transfer.

We hypothesized that this "asymmetry" effect was induced by the activation functions applied after each convolution layer inside the elu network and, therefore, carried out a deeper analysis of the effects of different operators. We considered three modifications of our network, namely, the use of exponential linear units (elu) [21], hyperbolic tangent (tanh) and bipolar exponential linear units (belu) [9] (Fig. 5). Due to the nature of the belu function, which doubles the output size, we had to adapt the number of convolutional filters in the first layer to comply with memory limits (Table I). Due to the properties of the belu non-linearity, this is not necessarily the same shape the preceding convolutional operator has. We denote that by writing 12*2 for e.g. the first convolutional layer conv_0 (Table I), meaning the convolutional operator outputs a tensor with 12 channels and next the belu unit doubles this to 24 channels.

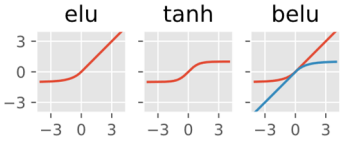


Fig. 5. Non-linearity functions used in the classification networks

In the following, we illustrate the effect of using different activation functions in neural architectures by applying them to a simple two-dimensional pixel image and a vertical edge detector filter (Fig. 6). We chose the commonly used rectified linear unit (relu) [23], [24] (9), additionally to elu [21] (10) and tanh (Fig. 5). The relu operator is implemented in the VGG19

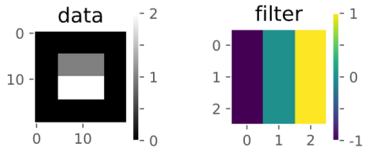


Fig. 6. Artificial data (left) and edge detector (right) for the activation function illustrations

[10] network as activation function used by Gatys et al. [1] for the original 2D Neural Style Transfer. Our voxel-based networks (Table I), use the elu layer, which is derived from the relu layer, and the tanh layer.

$$relu(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases} \tag{9}$$

$$elu(x) = \begin{cases} x & x > 0 \\ \alpha \cdot (e^x - 1) & x \leq 0 \end{cases} \tag{10}$$

In a classification network, a non-linear operator is usually applied on the activation map. We here receive the activation map conv2d from convoluting the filter across the data image with padding mode "same" (Fig. 7).

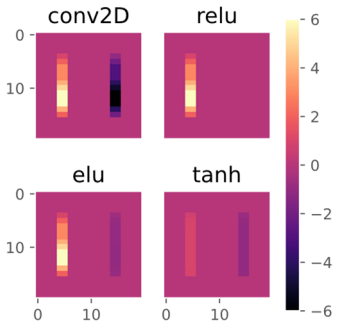


Fig. 7. Top Left: Activation map of the convolutional layer with the edge detector filter (Fig. 6). Top right, bottom left and right: Activation maps for relu, elu and tanh operators.

The responses of the chosen activation functions show fundamentally different behavior (Fig. 7). Only the tanh activation function returns a "symmetric" result, i.e. both vertical edges of the artificial 3D data are handled equally. On the contrary, the tanh function lacks the ability to "spike". Both, relu and elu activation function show "asymmetric" behavior, which we hypothesized to be responsible for the asymmetric effects on 3D data shown in Fig. 3 and Fig. 4. The information loss during the forward pass in the classification networks hinders a proper shape reconstruction during our optimizations. This is less severe for the elu layer because of the exponential part, which delivers "clipped" information for the right edge of our artificial data patch. However, since we perform an optimization, which propagates the gradients through the classification network back to our to be modified 3D voxel model, the derivatives of relu and elu also have a negative impact for shape reconstructions. Both activation functions have a derivative of 0 or close to zero for negative values, which abruptly stops the backwards flow of information through many connections in the used networks. The hyperbolic tangents shares the negative effect of its close to zero gradients for greater positive and negative values.

Eidnes and Nøkland [9] have introduced bipolar activation functions, effectively doubling the channel size of the network tensor,

$$f_B(x) = \begin{bmatrix} f(x) \\ -f(-x) \end{bmatrix}, \tag{11}$$

where $f(x)$ denotes any commonly used activation function like relu or elu.

The activation maps resulting from applying the belu operator (Fig. 8) combine the advantages of other operators, it allows for symmetric activations while showing the ability to spike during inference and backpropagation.

Hence, the belu operator is a promising candidate for resolving the asymmetry observed in 3D style reconstruction results. Including the operator in the network affects the network architecture because of the doubled channel size of its output tensor. Hence, the tunable variable amount of the following convolutional layer is also doubled. Style reconstruction experiments implementing the different operators in the classification network architectures following Table I are presented in Section IV.

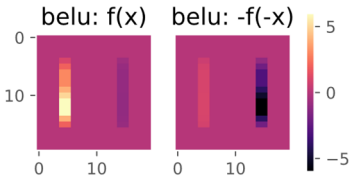


Fig. 8. Activation maps for the bipolar exponential linear unit (belu)

C. *Quantitative Evaluation of 3D Neural Style Transfer*

To evaluate the 3D style reconstruction quantitatively, we propose to use local 3D shape descriptors similar to descriptors introduced for the 2D domain or descriptors for 3D mesh representations [25]. We propose two types of surface shape descriptors (Fig. 11), a *slices* descriptor and a *neighborhood bin grid*. Both descriptor types are applied to every surface voxel, where the slices descriptor counts the number of voxels for six grid slices around a single surface voxel, while the neighborhood descriptor counts the number of voxels in the immediate neighborhood of the surface voxel. This results in two histograms describing the frequency of occupancy numbers over all surface voxels. In the case of the slices descriptor, we did not differentiate between the six sides and simply stacked the grid occupancy values in a single vector. Both histograms are normalized by the number of surface voxels to make the histograms less dependent on the global size of the voxel model. Normalization is important for the style reconstruction evaluation since we assume style to be independent of the total shape size. To evaluate the geometric similarity between two shapes, we calculated the distance between the histograms for two shapes. Here, we apply the earth-mover- or Wasserstein-

distance [26], the mean square error (mse) and the mean absolute error (mae). As a side note, even if we apply our shape descriptors in the present paper to assess the style reconstruction results, they are general methods and will likewise be applied in our future 3D Neural Style Transfer pipeline to quantify the quality of the generated designs.

IV. Experiments

In this section, we provide details on the performance of the three variants of the classification network with different activation functions, elu, tanh, and belu.

A. *Network Validation: Classification Performance on ModelNet40*

We trained our three variants of voxel classification networks (Table I) on the aligned ModelNet40 dataset. The triangular meshes were converted to 256 voxels per dimension (Fig. 1, right). During the training we augmented the data by jittering the models along each axis without rolling over expansion limits. For optimizations, we relied on the Adam [27] algorithm. The networks were trained for a fixed number of epochs. Performance scores are based on the evaluation of the ModelNet40 test set. We intentionally omitted a further split into training and test set for additional hyperparameter tuning since the focus here was not to optimize for a maximum classification performance. Rather, we wanted to make use of the full ModelNet 40 training set for training of the neural network that we assumed to be beneficial for the later usage of the filters during Neural Style Transfer. In Table II, we report a performance increase of 2% of our best 256^3 voxel models over our best 64^3 voxel model. This slight increase can be explained by the increase in geometric detail due to the higher resolution of the voxel models.

In contrast to the elu and belu variants, the tanh network showed a low performance, which coincides with observations reporting on the lack of performance of sigmoid and tanh activation functions compared to the family of activation functions similar to rectified linear units (relu) used in deep neural networks [23], [24]. Note that the focus of our network architecture development lies on improved usability of these networks in a later 3D style transfer optimization task, not maximum classification performance. Hence, we are interested in sufficiently high classification accuracy, while we did not aim for a further optimization of the architecture towards an optimal classification performance. We depict the accuracy of every epoch on the full training and testing dataset (Fig. 9). Both, elu and belu variants showed close to identical performance, while the tanh version performed worse over the whole training period. Especially, the test performance was highly unstable. For the following experiments, we chose the network variants with the highest test score for each architecture.

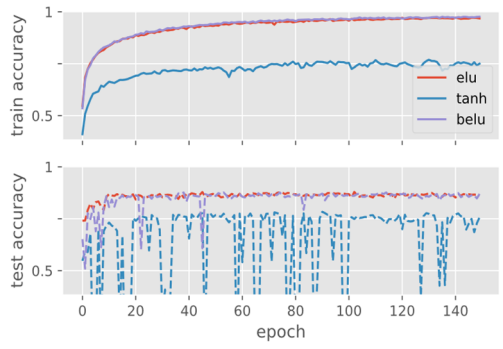


Fig. 9. Training and testing accuracy on ModelNet40 for the network architectures listed in Table I.

TABLE II
3D VOXEL CLASSIFICATION NETWORK PERFORMANCES

Classifier	ModelNet40 test set accuracy	Voxel data
VoxNet [28]	83%	32^3
ST_Enc_Pool_big [7]	86%	64^3, aligned
Net_elu	88%	256^3, aligned
Net_tanh	78%	256^3, aligned
Net_belu	88%	256^3, aligned

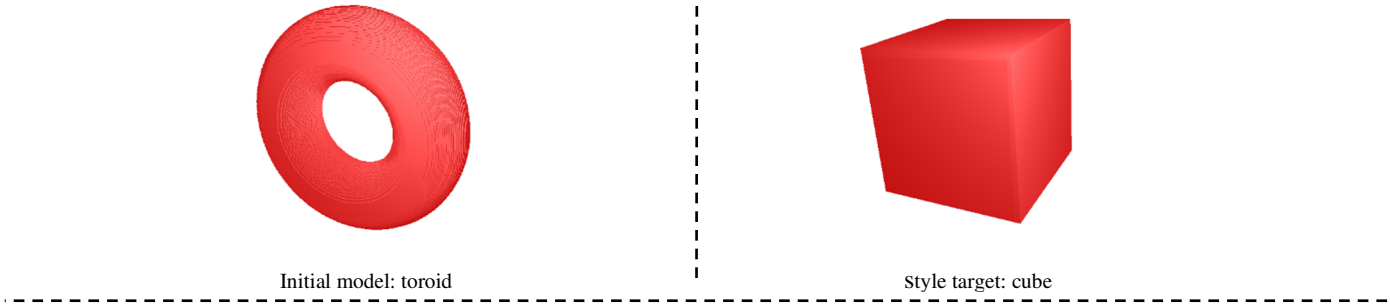
B. *3D Voxel Style Reconstructions*

In this section, we performed style reconstruction on 3D voxel models with a maximal edge length of 256 and compare the results for the three, elu, tanh, and belu network variants (Fig. 10). We took an initial voxel model as the to be optimized shape and a second voxel model as a style target. The initial model can be of arbitrary shape. Since we took a cube (Fig. 10, top right) as style target, we chose a toroid (Fig. 10, top left) as initial model because it does not share surface features with the cube and is thus assumed to allow for a more straightforward evaluation of the style transfer. Also, the global expansion of the toroid differs from the cube compared to e.g. a standard sphere. Based on (4) and the Adam optimizer, we backpropagated an update signal (2) for each possible voxel position in our 256^3 grid through our network variants to our initial model. Thus, the initial shape morphed to the outputs shown in rows 2-4 in Fig. 10. The learning rate was manually adapted for each setup and the best results are presented.

The tanh variant of our networks failed to produce a reasonable result, where the reconstructed model lacked all cubic visual features. However, this setup did not suffer from noise at the edge faces and corner regions of the available voxel grid. The elu variant generated occasional cubic features but with the asymmetric effects (section III.B), while suffering from the most noise, especially in the border region of the voxel grid. We cut off these regions for the sake of visualization. The belu variant produced the most promising style reconstruction with recognizable cubic features evenly distributed over the whole shape. The reconstructed style showed stair-like structures and a close-to-perfect rectangular shaped pillar, which indicates a proper reconstruction of the cubic style, where the explicit shape is of no importance but only the occurrence of style features. The observable asymmetry of the style features on the shape generated with the elu network is not present. The belu network also produced less noise in the border regions compared to the elu setup. In summary, the usage of belu activation function significantly improved the visual quality of the output shapes.

C. *Quantative Style Reconstruction Evaluation*

In addition to the subjective visual evaluation of the former section, we applied our proposed 3D shape descriptors to



Initial model: toroid

Style target: cube



a) elu style reconstruction front and back



b) tanh style reconstruction front and back



c) belu style reconstruction front and back

Fig. 10. First row: initial and target voxel model used for the style reconstruction optimization. Second to fourth rows: style reconstruction results for each network configuration depicted from two opposing viewpoints.

quantitively describe the improvement of the style reconstructions generated with our proposed belu classification network. Histograms are shown for the slices descriptor (Fig. 12) and for the neighborhood descriptor (Fig. 13). We see that both surface descriptor types and all distance measures align with the visual evaluation. Assessing the distance measure correlation for the cube and each network (2nd column in each matrix), we obtained the smallest distance between the reconstructed style and the style target for our new belu network (lighter colors indicate smaller distances). The tanh network performed worst behind the elu network. The toroid columns additionally confirm our intuitive findings since the distance of the belu network to the toroid is larger than the distance of the elu and tanh network but smaller than the cube's distance (section IV.B.). Nevertheless, a smaller difference between the belu network distance and the cube's distance to the toroid

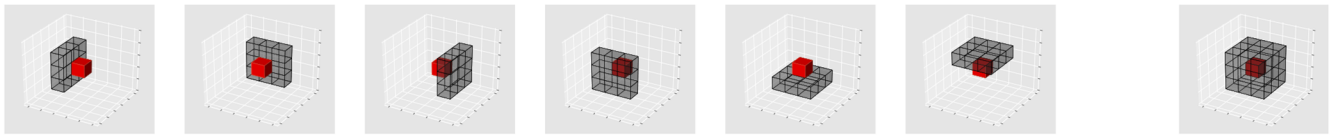


Fig. 11. The red cube denotes a surface voxel. Left: Slices descriptors for each side of the surface voxel. Right: Neighborhood descriptor for a surface voxel

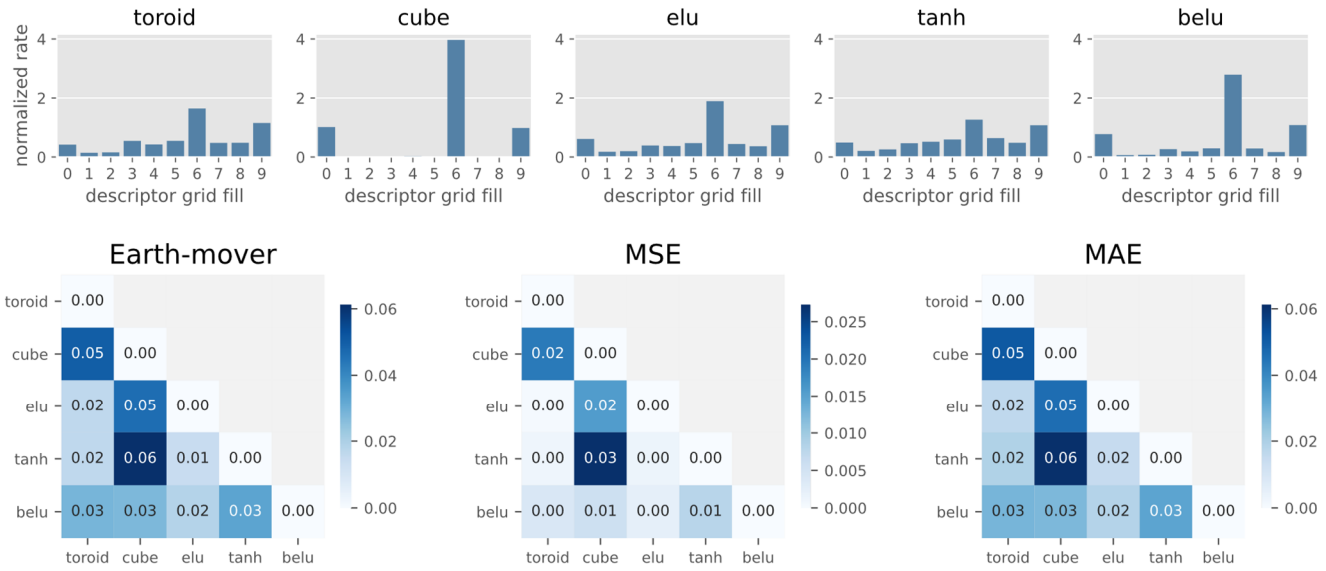


Fig. 12. Top: Histograms of the slices descriptor for each model depicted in Fig. 10. Bottom: Distance measure correlation matrices based on the slices histograms

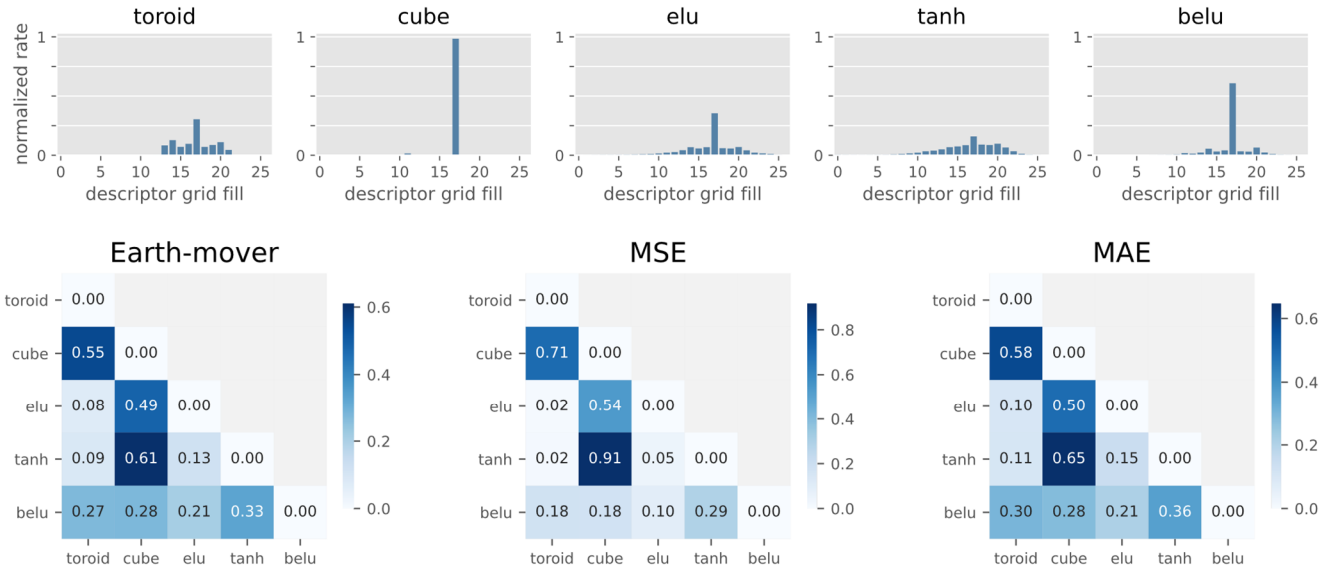


Fig. 13. Top: Histograms of the neighborhood descriptor for each model depicted in Fig. 10. Bottom: Distance measure correlation matrices based on the neighborhood histograms

would be desirable. This is potentially achievable by the further tuning of the optimization process by e.g. applying a denoising regularization like the total variation loss.

V. Conclusion

In the present paper, we further improved 3D voxel-based Neural Style Transfer by increasing the voxel resolution of every dimension by a factor of four. The resulting 256^3 networks outperformed our previous 64^3 voxel-based networks. By applying the bipolar exponential linear unit as activation function for the convolutional layers, we significantly improved the applicability of the classification networks for style reconstruction indicating essential advantages for 3D Neural Style Transfer. This was especially crucial because the number of convolutional filters that could be used in the architecture were limited by memory constraints. The new network architecture was successfully applied in a style reconstruction test. Here we showed superior reconstruction capabilities by visually inspecting the generated shapes and substantiated our subjective findings with the calculation of distance measures based on voxel surface descriptors that aligned with the visual impressions. To the best of our knowledge, this is the first attempt to quantitatively describe style similarity of voxel models using local 3D descriptors.

We showed again that the underlying classification network architecture for voxel-based 3D Neural Style Transfer is especially important compared to the two-dimensional image setup. First, today's computational limitations lead to smaller 3D networks with less filters in relation to 2D setups. Second, the lack of sufficiently large 3D datasets hinders the training of rich neural networks. Both issues imply a stronger vulnerability of the less robust voxel-based style transfer setups towards misconfigurations of the classification network. This work provides an additional, important step towards high-quality 3D voxel-based Neural Style Transfer.

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2414–2423, 2016.

[2] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song, "Neural style transfer: A review," CoRR, vol. abs/1705.04058, 2017.

[3] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang, "Universal style transfer via feature transforms," CoRR, vol. abs/1705.08086, 2017.

[4] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9906 LNCS, pp. 694–711, 2016.

[5] D. Holden, I. Habibie, I. Kusajima, and T. Komura, "Fast Neural Style Transfer for Motion Data," IEEE Computer Graphics and Applications, vol. 37, no. 4, pp. 42–49, 2017.

[6] T. Friedrich, N. Aulig, and S. Menzel, "On the Potential and Challenges of Neural Style Transfer for Three-Dimensional Shape Data," in EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization, pp. 581–592, 2019.

[7] T. Friedrich, and S. Menzel, „Standardization Of Gram Matrix For Improved 3D Neural Style Transfer," in Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1375–1382, 2019.

[8] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 07-12-June-2015, pp. 1912–1920, 2015.

[9] L. Eidnes, and A. Nøkland, "Shifting Mean Activation Towards Zero with Bipolar Activation Functions," arXiv:1709.04054, 2017.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proceedings of 3rd International Conference on Learning Representations (ICLR), 2015.

[11] C. Ma, H. Huang, A. Sheffer, E. Kalogerakis, and R. Wang, "Analogy-driven 3d style transfer," in Computer Graphics Forum, vol. 33, no. 2. Wiley Online Library, pp. 175–184, 2014.

[12] T. Friedrich, S. Schmitt, and S. Menzel, "Rapid Creation of Vehicle Line-Ups By Eigenspace Projections for Style Transfer," in Proceedings of the Design Society: DESIGN Conference, pp. 867–876, 2020.

[13] H. Kato, Y. Ushiku, and T. Harada, "Neural 3D Mesh Renderer," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3907–3916, 2018.

[14] B. Kim,V. C. Azevedo, M. Gross, and B. Solenthaler, "Transport-based neural style transfer for smoke simulations," in ACM Transactions on Graphics 38(6), 2019.

[15] D. Sýkora, O. Jamriška, J. Lu, and E. Shechtman, "StyleBlit: Fast Example-Based Stylization with Local Guidance," in Computer Graphics Forum, 38(2), pp. 83–91, 2018.

[16] Y. Ren, and H. Zheng, "The Spire of AI - Voxel-based 3D Neural Style Transfer," in Proceedings of the 25th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA), 2020.

[17] C. R. Qi, H. Su, M. Niebner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view CNNs for object classification on 3D data," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-December, pp. 5648–5656, 2016.

[18] "Understanding Formal Analysis.", [Online]. Available: http://www.getty.edu/education/teachers/building_lessons/formal_analysis.htm, [Accessed: 01- Jul- 2020]

[19] F. S. Nooruddin, and G. Turk, "Simplification and Repair of Polygonal Models Using Volumetric Techniques," in Proceedings of IEEE Transactions on Visualization and Computer Graphics 9(2), pp. 191–205, 2003

[20] P. Min, "binvox", [Online]. Available: http://www.patrickmin.com /binvox 2004-2020, [Accessed: 01- Jul- 2020]

[21] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in Proceedings of 4th International Conference on Learning Representations (ICLR), 2016.

[22] S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv:1502.03167, 2015.

[23] V. Nair, and G. E. Hinton,. "Rectified linear units improve Restricted Boltzmann machines," in Proceedings of the 27th International Conference on Machine Learning (ICML), pp. 807–814, 2010.

[24] X. Glorot, A. Bordes, and Y. Bengio,. "Deep sparse rectifier neural networks," in Journal of Machine Learning Research 15, pp. 315–323, 2011.

[25] G. Lara López, A. Peña Pérez Negrón, A. De Antonio Jiménez, J. Ramírez Rodríguez, and R. Imbert Paredes, "Comparative analysis of shape descriptors for 3D objects," in Multimedia Tools and Applications, 76(5), pp. 6993–7040, 2017.

[26] Y. Rubner, C. Tomasi, and L. J. Guibas, "A Metric for Distributions with Applications to Image Databases," in Proceedings of International Conference on Computer Vision (ICCV), pp. 59–66, 1998.

[27] D. P . Kingma, and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.

[28] D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," in Proceedings of IEEE International Conference on Intelligent Robots and Systems, vol. 2015-Decem, pp. 922–928, 2015.