ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2.
Available from http://www.i6doc.com/en/.

Visualizing and Improving 3D Mesh Segmentation with DeepView

Andreas Mazur[*], Isaac Roberts, David Leins,
Alexander Schulz and Barbara Hammer [†]

CITEC – Center for Cognitive Interaction Technology
Bielefeld University – Faculty of Technology
Inspiration 1, 33619 Bielefeld – Germany

Abstract. While 3D data is rich in information, it often comes with the drawback of being tedious to handle. Recent work in the Geometric Deep Learning community focused on developing high quality 3D datasets for tasks like mesh segmentation. However, the label quality can never be assured to be perfect. To improve label quality in 3D datasets, we propose an interactive algorithm combining DeepView, a method to visualize the classification function of neural networks, with Intrinsic Mesh CNNs, which generalize the convolution to Riemannian manifolds, to smartly select adequate sets of vertices from triangle mesh data for label correction.

1 Introduction

Deep learning models achieve state-of-the-art results in various tasks and research areas. For non-Euclidean data such as graphs and manifolds in tasks like 3D object segmentation, geometric deep learning (GDL) approaches have shown to be very promising [1]. Several articles have shown that explainable AI (xAI) methods can increase trust in machine learning models or even help to improve them by detecting artifacts in their training data [2, 3]. While some of them also target GDL-models, such as intrinsically interpretable methods [4, 5] and adaptations of post-hoc methods (see [5] for examples), most of them are limited to local interpretation methods specifying e.g. which parts of the current input are important for the resulting prediction. Global dimensionality reduction (DR) based methods do exist for other domains [3, 2, 6] and have been shown to be useful to detect different types of attacks, artifacts in the data and to understand model behaviour during training, but have not been investigated for GDL models. Methods from xAI can also be used to identify data points with incorrect ground truth labels. While there do exist classic approaches for classification in the presence of label noise [7], they usually work fully automated and, hence, are prone to errors. In contrast, recent work [8, 9] in the image segmentation domain, where coarse pixel classifications correspond to label noise, utilizes an

[*] Corresponding author: amazur@techfak.uni-bielefeld.de

[†] The authors gratefully acknowledge the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 520459685; from the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101073307 and from the state of NRW by the Ministry of Economic Affairs, Industry, Climate Protection and Energy (MWIKE) for the project "Hybrid Living" under the grant agreement No 005-2211-0002.

assisted-manual approach where automated labels are corrected pixel-wise by human annotators. While such pixel-wise annotation works well for image data, it becomes tedious for 3D segmentation problems with point clouds and triangle meshes [10]. In this work we adopt the DeepView method [3] to Intrinsic Mesh CNNs (IMCNNs) [11] by proposing a 3D segmentation-label correction algorithm for noisy datasets. We argue that by visualizing the classification space of an IMCNN, we are able to combine the model's learned knowledge with an oracle's knowledge and by that can select meaningful subsets of labels to correct. We show the algorithm's efficacy by improving segmentation labels of PartNet-Grasp, a subset of triangle meshes from the widely known PartNet dataset [10], which we make publicly available[1].

2 Background

DeepView [3] is a framework to visualize a part of the prediction function of a deep neural network classifier together with it's training or testing data. It consists of four core steps which include (i) projecting the data to two dimension using a discriminative (sometimes also referred to as supervised) DR method, (ii) sampling a regular grid in the 2D space and mapping it to the original input space, (iii) applying the classifier to the projected samples to obtain the predicted class label and the certainty estimate and finally (iv) visualizing the certainties and predicted classes in the background of the 2D scatter plot to obtain an approximation of the decision function. The DR step is the most influential one since it selects a subspace of the input for visualization. To this end, DeepView employs a discriminative variant of UMAP [12], that uses regular UMAP together with a discriminative distance metric that emphasizes directions in the data space, where classifier predictions change.

Intrinsic Mesh CNNs [11] generalize the standard Euclidean convolution to convolutions on manifolds:

$$(s * t)(\vec{u}) = \int_{B_R} t(\vec{v}) \; [s \circ \exp_{\vec{u}} \circ \omega_{\vec{u}}](\vec{v}) \; d\vec{v} \tag{1}$$

Hereby, $B_R \subset \mathbb{R}^n$ represents the sphere with radius R around $\vec{0}$, $t : B_R \to \mathbb{R}$ a trainable template, $s : M \to \mathbb{R}$ a signal function defined on a compact Riemannian manifold M, $\exp_{\vec{u}} : T_{\vec{u}}M \to M$ the exponential map for the tangent space at $\vec{u}$ and $\omega_{\vec{u}} : B_R \to T_{\vec{u}}M$ the selected gauge for $T_{\vec{u}}M$. The manifold convolution essentially defines a parameterization space for the local tangent space in the point of convolution and utilizes a pullback of the signal to said parameterization space. The pullback thereby respects the local geometry of the manifold. Eventually, the convolution is computed in the parameterization space. IMCNNs aim to learn *intrinsic properties* of the underlying manifold, which are invariant towards metric-preserving transformations of the manifold such as rotations and translations.

[1] Experiment code: `https://github.com/andreasMazur/VisMeshSegmentation`.

Algorithm 1 DeepView 3D Segmentation Label Correction

```
Require: Dataset $\mathcal{D} = \{(m_i, l_i)\}_{i=1}^{N}$ of $N$ meshes $m_i$ with vertex labels $l_i$
imcnn ← trainIMCNN($\mathcal{D}$)
$\tilde{\mathcal{D}} \leftarrow \emptyset$                                   ▷ $\tilde{\mathcal{D}}$ represents a new dataset
for $(m_i, l_i)$ in $\mathcal{D}$ do
    $\mathcal{I} \leftarrow$ DeepView(imcnn, $(m_i, l_i)$)          ▷ $\mathcal{I}$ represents the DeepView image
    $\mathcal{L} \leftarrow \emptyset$                               ▷ $\mathcal{L}$ represents the list of all corrections
    while oracle dissatisfied do
        $\mathcal{V} \leftarrow$ selectRegion($\mathcal{I}$)             ▷ select vertices $\mathcal{V}$ in $\mathcal{I}$ with lasso tool
        highlightIn3D($m_i, l_i, \mathcal{V}$)
        $\mathcal{C} \leftarrow$ proposeCorrections($\mathcal{V}$)        ▷ labels $\mathcal{C}$ for $\mathcal{V}$ chosen by the oracle
        $\mathcal{L} \leftarrow$ addToCorrections($\mathcal{L}$, $\mathcal{V}$, $\mathcal{C}$)    ▷ add new, replace old corrections
    end while
    $(\tilde{m}_i, \tilde{l}_i) \leftarrow$ applyCorrections($\mathcal{L}$, $(m_i, l_i)$)
    $\tilde{\mathcal{D}} \leftarrow$ addToDataset($\tilde{\mathcal{D}}, (\tilde{m}_i, \tilde{l}_i)$)
end for
return $\tilde{\mathcal{D}}$
```

3 Sampling from PartNet: PartNet-Grasp

In this work we evaluate our methodology on a subset of the PartNet dataset [10], which can be used to train segmentation models for the downstream task of e.g. object grasping. We select triangle meshes from the *Mug* class of the PartNet dataset, which consists of 192 meshes with annotations describing part elements such as "body" or "handle". While PartNet contains segmented regions as individual meshes, we are interested in single object meshes containing multiple segments, in which each vertex is assigned to exactly one segment. Since PartNet builds on top of ShapeNet [13] we can transfer the segment-meshes from PartNet to their corresponding original meshes in ShapeNet and assign PartNet labels to ShapeNet meshes using the following heuristic:

(i) We calculate the scaling factor between the largest dimension of the original mesh in ShapeNet and the corresponding segment meshes from PartNet that in combination build the original ShapeNet mesh. (ii) The translation error is calculated as the difference between the centers of mass of the original ShapeNet and the scaled combined mesh. (iii) The rotation is either correct or off by $-90°$ around the z-axis. By comparing the Chamfer-distance between the vertices of both meshes, we determine which rotation results in a better alignment. (iv) For each vertex in the ShapeNet mesh we calculate the signed distance to each PartNet mesh and assign the segment-label with the smallest distance.

For the downstream task of classifying graspable regions, only the *handle* annotations are retained and the remaining vertices are assigned the *body* class, i.e., non-graspable regions. Mugs containing liquids or other additional objects are excluded from the dataset. The final dataset consists of 100 meshes, ranging from 74 - 8848 vertices, with an average of 1504 vertices per mesh. Since the focus of this subset lies on labeling graspable regions of mugs we refer to it as *PartNet-Grasp* in the course of this work.

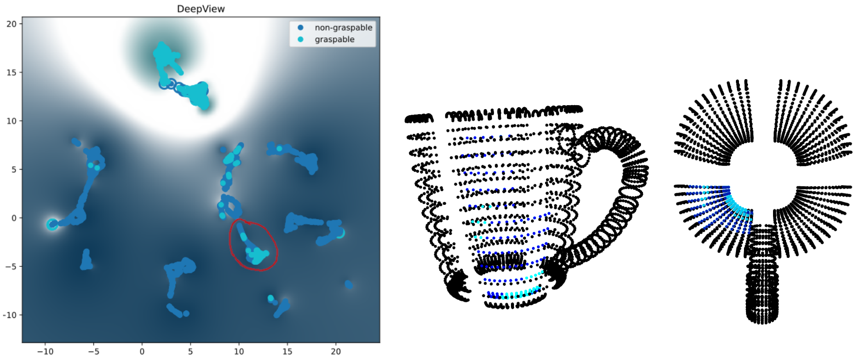

Fig. 1: **[Left]** DeepView embedding showing 2D projections of the data points corresponding to points of the 3D mesh together with a projection of the classification function in the background. The selected area of vertices via the matplotlib lasso tool for 3D visualization and potential label correction is depicted in red. **[Right]** The encircled vertices highlighted on the 3D shape of PartNet-Grasp. The oracle can now determine whether the labels are correct.

4 Segmentation Label Correction Algorithm

As the labels in PartNet-Grasp were computed using a heuristic, optimal label quality is not guaranteed. To improve the label quality, we propose a novel method for correcting 3D segmentation labels using DeepView and IMCNNs. Our method starts by training an IMCNN on the available dataset that contains noisy labels for 3D mesh segmentation. We then use the trained IMCNN to iteratively compute DeepView visualizations per mesh. Thereby, the DeepView image portraits not only the projected classification function of the IMCNN in the background but also 2D vertex-feature projections of the vertices from the current mesh. Using the projected classification function and 2D vertex-feature projections for assistance, an oracle selects a set of points within the DeepView image, subsequently receives a 3D visualization of the selected points highlighted on the shape and eventually corrects the labels if it determines that they have been assigned to the wrong class. This process is repeated until the oracle does not see any other incorrect labels for the current mesh. It receives new meshes from the dataset for correction until the labels for all meshes in the dataset have been corrected. Algorithm 1 and Figure 1 describe the process in more detail.

5 Experimental Evaluation

We first evaluate whether training a classifier on the corrected samples yields a significant improvement, and second, compare to a label noise detection baseline.

Impact of Label Correction in Classification We conduct 30 IMCNN training

Network Architecture	Labels	Accuracy	Loss
$\mathrm{BN}(\mathrm{FC}(3 \to 64))$	Uncorrected	0.86 ± 0.02	0.32 ± 0.02
$\mathrm{BN}(\mathrm{ISC}(64 \to 8, 96))$	Corrected	0.94 ± 0.01	0.21 ± 0.01
$\mathrm{AMP}(8, 96 \to 96)$	**Improvement**	**0.08**	**0.11** (34%)
$\mathrm{FC}(96 \to 2)$	WRT p-value	$< 10^{-10}$	$< 10^{-10}$

Table 1: **[Left]** The IMCNN-architecture used in our experiments. The inputs are 3D coordinates of the mesh vertices; *FC* stands for *fully connected*, *BN* for batch normalization, ISC for *intrinsic surface convolution* (cf. Eq. 1) and AMP for *angular max-pooling* [11]. We apply ReLU-activation functions after each layer except the last one. The network shall predict whether a vertex belongs to a graspable mesh segment. **[Right]** Mean test accuracies and losses as well as their standard deviations over 30 training runs for the corrected and uncorrected PartNet-Grasp datasets. The samples for the WRT are given by the test accuracy sets or test loss sets, respectively.

runs on the corrected as well as uncorrected PartNet-Grasp dataset, i.e. 60 training runs in total, using the architecture from Table 1 on the left. We split the corrected and uncorrected version of PartNet-Grasp equally into a 70/10/20 training, validation and test split and test all models on the corrected test dataset. We use the collected test accuracies and losses to proceed with a two-tailed Wilcoxon rank-sum hypothesis test (WRT). This test examines the null-hypothesis whether our collected accuracies or losses, respectively, arise from the same distribution.

In our case, we compare the test statistics of the models trained on the corrected versus uncorrected datasets. The averaged test accuracies, test losses, their variances and the p-values for the WRT can be seen in Table 1 on the right. Since both p-values are smaller than $\alpha/2$ for a significance level of $\alpha = 0.05$ it becomes evident that we can reject the null-hypothesis. We thus can observe a statistical significant difference between the collected samples for corrected and uncorrected data. The difference is given by an improvement of 8% in test accuracy and 48% in test loss on average.

Comparison to other Label Noise Detection Methods We additionally compare our algorithm to a simple filter approach [7], which selects those points that are misclassified by the model with high certainty. Here we compute how many of the points selected by our strategy would be retrieved with the filter method, by simply looking at all misclassified points. Thereby, we avoid dealing with multiple thresholds to specify high certainty. We obtain an average recall of 0.63 ± 0.40 and an average precision of 0.39 ± 0.33, averaging over all meshes. In total, more then 30% of the incorrectly labeled points could not be retrieved this way on average, while suggesting many points for relabelling, which the oracle did not deem incorrectly labeled.

6 Conclusion

In this work we have proposed a segmentation label correction algorithm that combines DeepView with IMCNNs to correct noisy segmentation labels in 3D datasets. We experimentally evaluated our algorithm at the hand of PartNet-Grasp, a sub-dataset we have sampled from PartNet, by conducting the Wilcoxon rank-sum test and compared it to a classical filter approach. Our experiment results showcase the algorithm's effectiveness.

References

[1] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE CVPR*, pages 5115–5124, 2017.

[2] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019.

[3] Alexander Schulz, Fabian Hinder, and Barbara Hammer. Deepview: Visualizing classification boundaries of deep neural networks as scatter plots using discriminative dimensionality reduction. In Christian Bessiere, editor, *Proceedings of IJCAI-20*, pages 2305–2311, 7 2020. Main track.

[4] Siqi Miao, Mia Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *ICML*, pages 15524–15543. PMLR, 2022.

[5] Siqi Miao, Yunan Luo, Mia Liu, and Pan Li. Interpretable geometric deep learning via learnable randomness injection. In *ICLR*, 2022.

[6] Xianglin Yang, Yun Lin, Ruofan Liu, Zhenfeng He, Chao Wang, Jin Song Dong, and Hong Mei. Deepvisualinsight: Time-travelling visualization for spatio-temporal causality of deep classification training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5359–5366, 2022.

[7] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF ICCV*, pages 4015–4026, 2023.

[9] Hoyoung Kim, Sehyun Hwang, Suha Kwak, and Jungseul Ok. Active label correction for semantic segmentation with foundation models. *arXiv preprint arXiv:2403.10820*, 2024.

[10] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of CVPR*, pages 909–918, 2019.

[11] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[12] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[13] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.