Point Cloud Classification with ModelNet40: What is left?

Jarne Van den Herrewegen [1 2] **Tom Tourwé** [1] **Francis wyffels** [2]

Abstract

State-of-the-art 3D classification models are showing saturating performance on the popular ModelNet40 benchmark. We investigate possible causes for the remaining mistakes and find various data-related issues. In summary, our goal is 1) to give suggestions for future dataset creation in 3D deep learning and 2) to provide ground-truth information on mistakes for evaluation of (future) automated data cleaning methods.

1. Introduction & Related Work

In the field of 3D deep learning, ModelNet40 (Wu et al., 2015) has been the most popular benchmark for 3D point cloud classification (4800 citations), amongst other important benchmarks such as ScanObjectNN (Uy et al., 2019) (400 citations) and ShapeNetCore55 (Chang et al., 2015) (3900 citations). Wu et al. released ModelNet40 in 2015 and reported 77.32% test accuracy with a 3D convolutional deep belief network. In the same year, Su et al. (2015) achieved 90.1% using a multi-view convolutional neural network. Point cloud classifiers caught up in 2017 when Qi et al. (2017) achieved 91.9% test accuracy, presenting PointNet++.

In recent years, however, new point cloud classifiers did not advance at the same pace on the ModelNet40 benchmark and seemingly approach an upper bound of 95%. At the same time, the performance on the ScanObjectNN classification benchmark has improved significantly, which indicates that modeling approaches for point cloud classification are advancing. We refer to Chen et al. (Chen et al., 2023), Table 1, for a comparative overview of several classification models evaluated on ModelNet40 and ScanObjectNN.

[1]Oqton AI, Ghent, Belgium [2]IDLAB-AIRO, Ghent University, Ghent, Belgium. Correspondence to: Jarne Van den Herrewegen <jarne.vandenherrewegen@oqton.com>.

Accepted as extended abstract to the Data-centric Machine Learning Research Workshop at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. *This work was supported by the Flemish Institute for Innovation and Entrepreneurship (VLAIO), grant number HBC.2022.0165.*

What is left in the final 5% of ModelNet40's test set? As mentioned in the DataPerf paper (Mazumder et al., 2022), it is important to inspect the mispredictions of a model to identify open challenges and to move forward within a subfield. Recently, Wang et al. (2023) investigated label errors in ModelNet40's test set. Human annotators agreed that 1.34% of the test set was incorrectly labeled. However, the identified label errors were not released to the public and the authors did not analyze other causes for mispredictions.

In this work, we report on more extensive investigation of errors[1] in the ModelNet40 test set. Among these issues, there is data duplication, data corruption, label errors, orientation misalignment and scale inconsistency. In summary, our goal is 1) to give suggestions for future dataset creation in 3D deep learning and 2) to provide ground-truth information on mistakes for evaluation of (future) automated data cleaning methods (e.g. data deduplication).

This report is part of ongoing research that would ideally lead to insight in the common problems of existing datasets and lead to a new, well-crafted grand challenge for 3D classification. We are looking for community feedback on our preliminary results and we welcome suggestions towards improved methodology and automation.

2. Analysis

The analysis of the ModelNet40 dataset is discussed in a step-by-step manner. The first step establishes a baseline with a PointNeXt-S (Qian et al., 2022) model on the original training set and test set. During the following steps, the model architecture remains fixed but the training set and the test set can be changed. For now, we focus on cleaning the test set in a semi-automated manner under the assumption that the training set can be cleaned in an automated manner once a golden standard has been established.

2.1. Step 0: Baseline

A baseline is set with the PointNeXt-S architecture. It provides state-of-the-art performance and comes with a well-maintained implementation. For inference and for training, we use ModelNet40 point clouds provided by the Stanford

[1]A list of errors is available at https://github.com/oqton/M40-cleaning.

University ShapeNet website[2] as done by Qian et al. (2022). The point clouds contain 2048 points, are scaled to the unit sphere and are subsampled to 1024 points during training. The test accuracy of the baseline model is 93.3%. The training set and the test set are referred to as *train v0* and *test v0*.

2.2. Step 1: Deduplication

While visually inspecting the dataset, various (near-) duplicates were noticed. Duplicates add redundant information and introduce data leakage between the test and training set. Although the ModelNet40 authors mention duplicate removal in their work, they do not specify the exact method. To deduplicate the full dataset, we execute the following procedure. First, embeddings for all objects are inferred with the baseline classifier backbone. Then, all pairs of objects for which the embeddings are closer than 0.01 cosine distance are considered duplicate candidates. The final candidate pairs are visually compared and annotated by a human. If a candidate pair is labeled as duplicate, one of the two objects is removed from the dataset. Objects from the test set are favored to remain in the dataset. In total, 35 test samples and 490 training samples are removed from the dataset, which is 4.26% of the full dataset. The *airplane* class in particular contained many duplicates (8 test samples, 151 training samples). The baseline classifier achieves 93.2% test accuracy on the deduplicated test set. Next, we retrain the model on the deduplicated training set, which we cal *train v1*. Because there is less data leakage, the performance drops to 92.9% on *test v0* and 92.8% on the deduplicated test set, referred to as *test v1*. We would like to note that there are still many near-duplicates in the dataset, causing non-negligible redundancy. The duplicate labels are ready to be released to the public for evaluating fully-automatic deduplication methods.

2.3. Step 2: Addressing data corruption

From visualizing the point clouds and their original meshes in the test set, two common types of data corruption were noticed: 1) there are flat objects and 2) some samples contain complete scenes instead of one object. To find flat objects, we propose a semi-automated solution by first calculating the bounding box for all test samples. We consider objects where the longest and the shortest dimension differ by a factor larger than 20 as flat object candidates (66 objects). We identify 7 test samples (3 *plant*, 4 *person*) as truly flat. The scenes (6 objects) were found manually while inspecting the test set. Removing these corrupted samples (*test v2*) changes test accuracy for the model trained on deduplicated data (*train v1*) from 92.8% to 92.9%.

[2] shapenet.cs.stanford.edu/media/modelnet40_ply_hdf5_2048.zip

2.4. Step 3: Label errors

Looking at the confusion matrix in Figure 1, we see that most errors made by the model trained on deduplicated training data are concentrated around a few classes: *flower_pot*, *plant*, *vase*, *table*, *night_stand* and *dresser*. We inspect all test samples – without knowing the models prediction – in these classes and agree that the bounds on some classes are not clearly defined. Out of 492 samples, we relabel 31 samples from these classes under the definitions in Appendix B. We also remove 6 samples that do not resemble any class in ModelNet40. After relabeling (*test v3*), the model trained on deduplicated data jumps from 92.9% to 93.5% test accuracy. Extending and improving our relabeling methodology is an ongoing effort.

2.5. Step 4: Alignment

The alignment in ModelNet40, provided by (Sedaghat et al., 2016), puts objects in a per-class canonical pose. The per-class alignment makes it easier to discriminate between classes. For example, if the classifier is trained with rotation augmentation, the performance drops to 91.4% (from 93.5% on *test v3*). The provided alignment is not perfect however: in the *tv_stand* class, we noticed several samples which are 90 degrees off the canonical pose. By rotating the mispredicted *tv_stand* samples by 90 degrees around the y-axis, 3 out of 6 previously mispredicted samples were correctly classified. This adjustment raises the performance of the model trained on deduplicated data (*train v1*) from 93.5% (*test v3*) to 93.6% (*test v4*).

2.6. Suggestion: Scale Inconsistency

Object size information could provide a discriminative feature for certain confusing classes (e.g. *cup* versus *vase*, *night_stand* versus *dresser*). However, most modeling approaches (including ours) use point clouds normalized to the unit sphere to reduce variance and do not make use of the scale information. A small visualization of objects from various classes, see Figure 2, indicates that the scale information could be useful. However, a histogram of object sizes in *night_stand* and *dresser*, see Figure 3, reveals that object sizes in the same class can differ by a factor of 10 or more, which is not in line with physical reality. We would like to highlight this issue towards future data acquisition.

3. Conclusion & Future Work

In this work, we took a closer look into the data to find causes for mispredictions in the ModelNet40 benchmark. During our analysis, we found data duplication, mesh corruption, label errors, alignment issues and scale inconsistencies. Various aspects of this research could be improved in future work. First, we could improve the automation and

the rigorousness of our current approaches. Additionally, we would like to inspect and clean the training to improve model performance. Finally, we should test the findings and methodology on additional datasets to validate their effectiveness and to discover other data issues.

References

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Chen, G., Wang, M., Yang, Y., Yu, K., Yuan, L., and Yue, Y. Pointgpt: Auto-regressively generative pre-training from point clouds. *arXiv preprint arXiv:2305.11487*, 2023.

Mazumder, M., Banbury, C., Yao, X., Karlaš, B., Rojas, W. G., Diamos, S., Diamos, G., He, L., Kiela, D., Jurado, D., et al. Dataperf: Benchmarks for data-centric ai development. *arXiv preprint arXiv:2207.10062*, 2022.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., and Ghanem, B. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022.

Sedaghat, N., Zolfaghari, M., Amiri, E., and Brox, T. Orientation-boosted voxel nets for 3d object recognition. *arXiv preprint arXiv:1604.03351*, 2016.

Su, H., Maji, S., Kalogerakis, E., and Learned-Miller, E. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 945–953, 2015.

Uy, M. A., Pham, Q.-H., Hua, B.-S., Nguyen, T., and Yeung, S.-K. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1588–1597, 2019.

Wang, J., Wang, R., Kim, T. S., Kim, J.-S., and Lee, H.-J. Diagnose label errors for 3d object classification. In *2023 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–4. IEEE, 2023.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

A. Confusion matrix

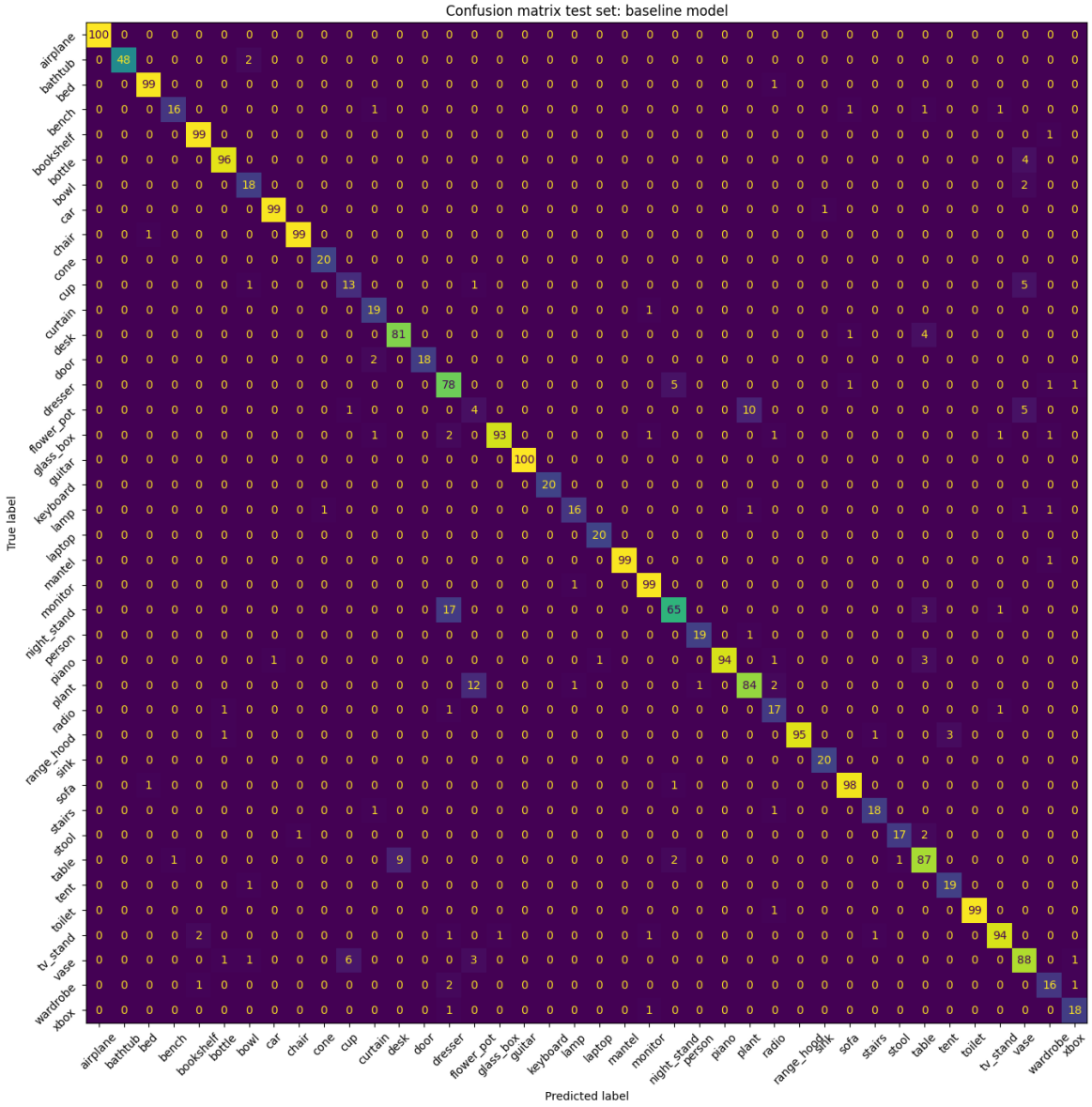

Figure 1. Confusion matrix for test set (*test v2* predictions made by the model trained on deduplicated data (*train v1* and after corrupt data removal. The most common errors occur for the classes *flower_pot*, *plant*, *night_stand*, *dresser*, *table* and *desk*.

B. Label error discussion

The samples were relabeled under the following definitions. All samples in *flower_pot* and *plant* containing flower petals are considered as *flower_pot* and other plant-like samples with only leaves are relabeled as *plant*. All samples in *flower_pot* which only show an empty pot are relabeled as *vase*. Several *table* samples are relabeled as *desk* following the definition saying that a table designed for only one person to sit at, is a desk. This relabeling methodology could be improved (e.g. to involve more annotators) and is subject to change for future work. However, we think this is an improvement over the original annotation strategy.

The original ModelNet40 dataset was labeled by asking annotators whether a shown object belonged to a given class, yes or

no. We argue that this practice ultimately leads to confusion between classes like *plant* and *flower_pot*. For example, if one is asked "Is this a plant?", showing a flower with many leaves in a pot, is tempting to say yes (in particular if one does not yet know about the *flower_pot* class.

C. Scale information

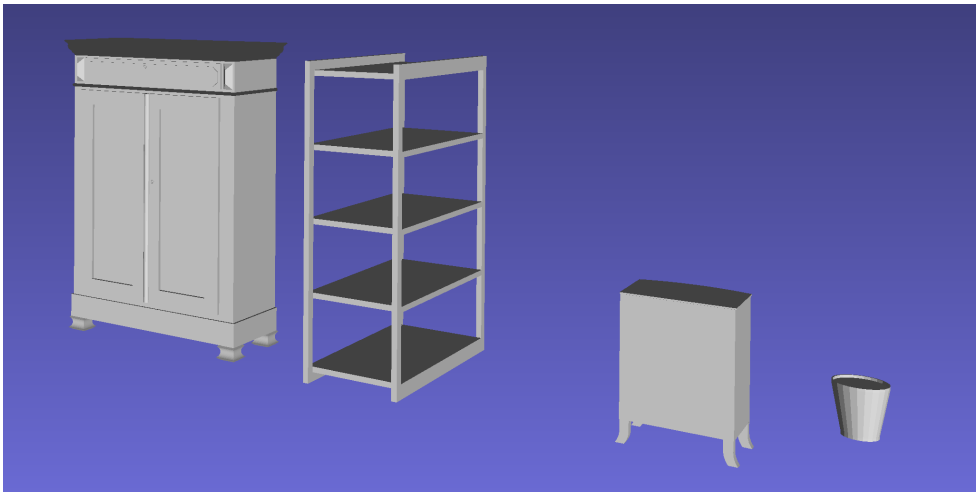

Figure 2. Example of a size comparison. All objects appear to have a relative scale that corresponds with realistic proportions.

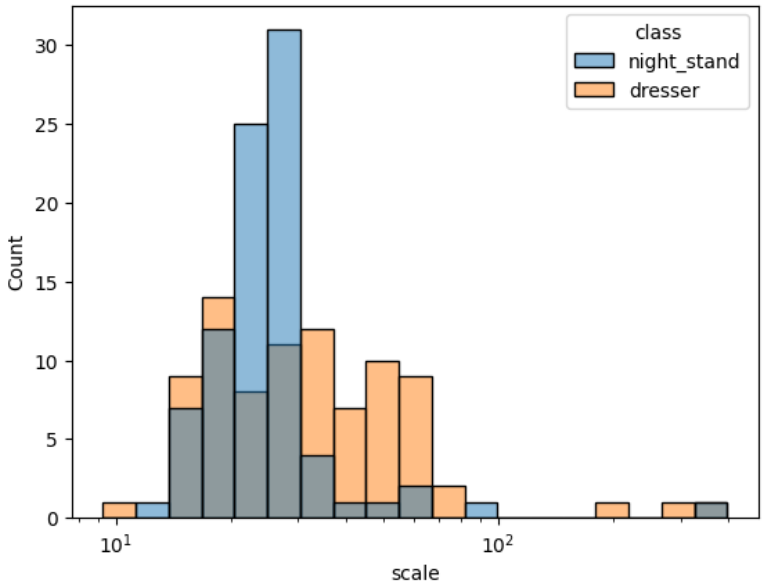

Figure 3. Histogram of object sizes in the classes *night_stand* and *dresser* in the test set of ModelNet40.