Learning Behaviour of Sparse Point-Voxel Convolution: Semantic Segmentation of Railway LiDAR scans

JASPER VAN DER WERF, University of Twente, The Netherlands

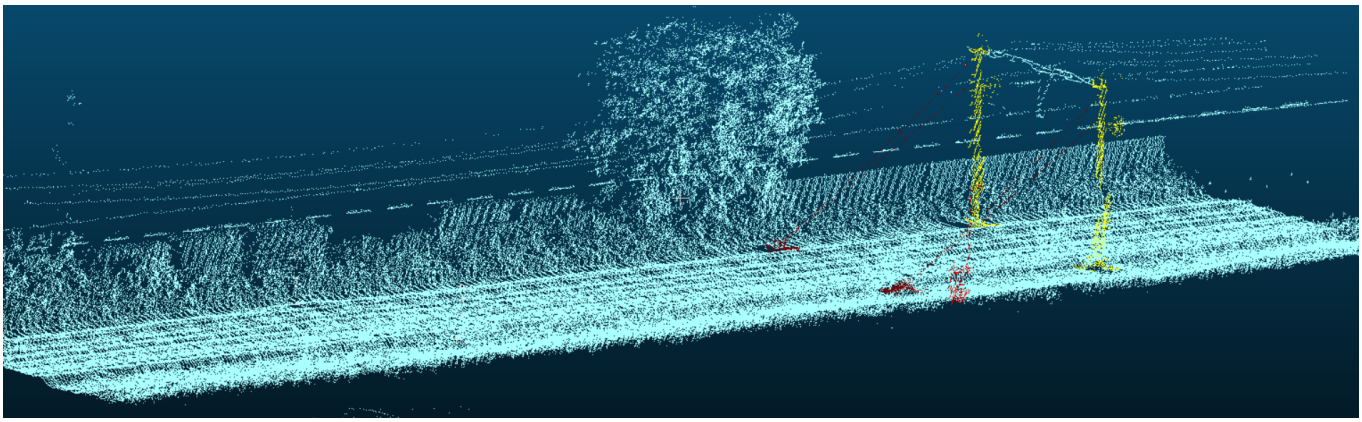

Fig. 1. A labeled point-cloud scene from LiDAR scans of the Dutch Railways

Supervised learning approaches require the creation of big datasets without upfront knowledge of the performance on these datasets. The high cost associated with such datasets highlights importance of being able to make performance estimates for full datasets through the analysis of learning behaviour on smaller datasets. This work analyzes learning behaviour with respect to variations in dataset size through a comparison of per-class intersection-over-union (IoU) against point- and scene-count in training data. In total, SPVConv models are trained for semantic segmentation of railways on various dataset-sizes. Linear regressions are extrapolated for the upward-trending performance on test data against the downward-trending performance on training data for both scene- and point-count, resulting in per-class predictions of IoU at their intersections. This work shows that some of the seen variations in IoU between the classes is very likely caused by a big class-imbalance in the dataset; this correlation is seen on limited data but also holds as the amount of data increases. In addition to the class imbalance, there are additional class-intrinsic factors that impact learning rate and IoU, shown through differences in slope for the various classes.

Additional Key Words and Phrases: Point cloud, railway, semantic segmentation, learning curve, class imbalance

1 INTRODUCTION

With the advent of PointNet by Qi et al. in 2017 [13], research on neural networks (NNs) for object recognition and semantic segmentation of point-clouds has seen tremendous progress [20]. These NNs can take an unstructured point-cloud as input and segment points or classify objects in a scene by using various localized feature-extraction techniques [4]. Supervised learning, the prevailing approach for training these models, necessitates the creation of substantial datasets in the form of labeled point-clouds. However, the creation of these large datasets is an onerous and costly task, and more importantly, it is undertaken without prior knowledge about the performance outcomes of fully trained models [3]. Given these constraints, it is paramount to be able to analyze the learning behavior of NNs on smaller, more manageable datasets. This would allow the creation of predictions about the potential performance on a full dataset, thereby circumventing the immediate need for the sizable investment involved in creating a comprehensive dataset.

Generally, as the volume of training data grows, performance on unseen data tends to improve, with an accompanying reduction in performance and overfitting on training data [12]. However, the performance improvements begin to diminish as the dataset size grows, approaching a limit when the model reaches its maximum performance. When provided with infinite training data, performance on training and testing data converges towards this upper limit. This relation of model-performance to data-availability is commonly known as the learning curve (with respect to data size) of a model, and is a metric frequently employed in performance analysis of machine learning models [10]. In this study, we focus on learning curves associated with data size, but extend the analysis to differentiate between the number of scenes (scene-count) and the number of class-specific data points (point-count) in training data.

Semantic segmentation and object recognition of railways have many important applications, predominantly in aiding the detection and assessment of railway infrastructure for inspection purposes [11, 15, 17, 18]. As part of a project commissioned by Strukton Rail, the Ambient Intelligence Lectorate (AMI) at Saxion University of Applied Sciences has annotated a LiDAR dataset of the Dutch railways. An exemplary scene from this dataset is depicted in Figure

TScIT 37, July 8, 2022, Enschede, The Netherlands

1, where unlabeled points are illustrated in blue, poles in yellow, and tension rods in red. The dataset employed in this research comprises 124 such scenes from two distinct railway lines in The Netherlands. Given its limited data volume, this dataset offers an ideal opportunity for the analysis and extrapolation of learning behavior.

In this study, we examine the response of the SPVConv [16] NN to variations in training data, through a comparison of each class's intersection-over-union (IoU) against the scene- and point-count of randomly-sampled subsets from the aforementioned dataset. A total of 38 SPVConv models are systematically trained for the task of semantic segmentation of railways, and each of these models is subsequently tested on two distinct test sets to determine class-specific IoU metrics. Through this investigation, we aim to answer the following research questions about the dataset and NN:

RQ1 *What is the impact of scene- and point-count in training data on the performance of semantic segmentation?*
RQ2 *How can learning curves relative to scene- and point-count be used to extrapolate class-specific IoU metrics to bigger datasets?*

2 RELATED WORK

The subsequent sections will cover some key topics related to the study of learning behavior in point-cloud semantic segmentation. We initially explore the landscape of neural network architectures, looking at various types of NNs used in semantic segmentation and their unique characteristics. Then, we'll give a brief rundown of established learning curve theory, to provide a theoretical background for our analysis. Lastly, we'll talk about class imbalances, which are especially important in our dataset, highlight some issues they might cause, and present some possible mitigations for these issues. This overview will help set the stage for the research that follows.

2.1 Semantic Segmentation of Point Clouds

Achieving state-of-the-art results for semantic segmentation and object detection of point clouds with NNs that directly targeted higher-dimensional data proved challenging prior to 2017 [4, 20]. However, the landscape was significantly transformed following the introduction of Qi et al.'s research [13] with their architecture, PointNet, designed to work directly with unstructured point-cloud data. Subsequent research has incorporated and improved techniques such as voxelization, point ordering, and feature fusion, enhancing training speed, improving scaling with larger point clouds, and reducing error rates [20].

One such recent advancement is the SPVConv neural network introduced in 2020, which applies voxelization and demonstrates "8x computation reduction and 3x measured speedup with higher accuracy" [16] compared to its predecessor, MinkowskiNet [2], on the widely recognized SemanticKITTI dataset [1]. At the time of writing, SPVConv ranks 7th on the SemanticKITTI leaderboard with a mean IoU (mIoU) of 66.4%. The top-performing model has an mIoU of 74.8%.

Neural networks of this type spatially partition the point-cloud into volumetric voxels, extracting features from these voxels to segment and classify the entire space. This method can pose challenges when objects within the scene exhibit significant size variation, as it requires the selection of a single voxel size as a model parameter. The SPVConv architecture addresses some of these challenges through its use of sparse point-voxel convolution, enabling more effective classification of both small and large objects within the same scene [16].

2.2 Learning Curve

Learning curves, which plot performance against an independent variable, provide insights into learning behavior. In the field of machine learning, data size, learning iterations, and neural network (NN) size care commonly used as the independent variable. However, theoretical exploration of learning curves with respect to data size remains relatively uncharted territory, still lacking a comprehensive mathematical foundation [7]. Recent years have seen increased efforts to establish this mathematical basis, with significant contributions from OpenAI [5] and Baidu [6].

These studies examine model performance as a function of varying data size, n, and propose a power-law relationship, $\epsilon(n) \propto \alpha m^{\beta_g}$, between data size n and generalization error ϵ, where α is a constant property of the problem. The scaling exponent β_g, which lies between 0 and -1, determines the learning curve's steepness, i.e. the rate at which a model family learns from additional training samples. For larger neural networks, values of β_g typically range from -0.07 to -0.35 [5, 6, 10].

The data size n is obtained as independent variable by randomly sampling subsets (scenes) of size n from the full dataset. This paper further differentiates data size into scene-count and point-count, to allow a more nuanced analysis of learning curves on a class-by-class basis. Scene-count in this work corresponds to data size in existing literature, whereas, to the best of our knowledge, no work has been done on learning curves specific to point-count. The point-count for a class is defined as the amount of that class's points in the training data and offers an alternative approximation of model performance relative to data size than scene-count, which does not consider variations in class balance within the training data.

2.3 Class Imbalances

Machine learning methods often rely on labeled datasets for training, and these datasets frequently exhibit unequal class distributions. A class imbalance occurs when one class significantly outnumbers another, leading to majority and minority classes [9]. Such imbalances can result in an overclassification of majority classes and an underclassification of minority classes - a problem particularly impactful when minority classes are of greater relevance than majority classes [8].

In their meta-study, Johnson et al. identify three strategies for addressing class imbalances: data-level methods, algorithm-level methods, and hybrid methods [9]. Data-level methods focus on modifying training data prior to feeding it into the model, employing techniques that oversample minority classes or interpolate different scenes to mitigate the imbalance. Algorithm-level methods, on the other hand, are applied after the data has been introduced to the model and include balanced loss functions, cost-sensitive learning techniques, and threshold adjustments. Hybrid methods combine

Label	Description	Occurence
0	Unlabeled	97.40%
1	Pole	2.24%
2	Tension rod	0.13%
3	Signal	0.15%
4	Relay cabinet	0.08%

Table 1. Class distribution

Set	Train	Test	Validation
The Veluwe	70 (81.4%)	8 (9.3%)	8 (9.3%)
Dronten	32 (84.2%)	6 (15.8%)	0 (0.0%)

Table 2. Train, test and validation splits

Size (scenes)	20	30	40	50	60	70	80	90	100
Repititions	10x	6x	5x	4x	3x	3x	3x	2x	2x

Table 3. Sizes and repititions of train partitions

data-level and algorithm-level approaches into a comprehensive solution.

3 METHODOLOGY

The following section outlines the methodology followed during our research to ensure the reproducibility of our findings. We begin by detailing the steps involved in pre-processing of the dataset to prepare it for semantic segmentation. This is followed by an explanation of our data partitioning strategy, outlining how we segmented our data into training sets for the various models. Subsequently, we delve into the specifics of our training process, focusing on our use of the SPVConv architecture and the specific parameters used. Lastly, we outline our systematic testing approach, explaining how we evaluated each model's performance across multiple test sets.

3.1 Data pre-processing

The AMI from the Saxion University of Applied Sciences created a high-resolution LiDAR dataset of railway tracks in The Netherlands, provided by Strukton Rail. This dataset covers two locations: one in The Veluwe and the other near Dronten. The AMI labeled the points within these datasets into five categories, as detailed in Table 1. The ground was then removed using the ground-removal method specified by Zermas et al. [19]. Following this, each location was broken down into scenes, each 75 meters long and 30 meters across.

The pre-processed dataset comprises 86 scenes from The Veluwe and 38 scenes from Dronten, totaling 124 scenes. Each scene contains approximately 100,000 to 300,000 points. An example of one of these scenes is shown in Figure 1. Table 1 illustrates a significant class imbalance in the dataset, even after ground removal. Over 97% of the data is composed of unlabeled points, while poles constitute just over 2%. The remaining three classes collectively make up less than 0.5% of the data.

3.2 Data partitioning

Before training any models, we split all scenes into training, testing, and validation sets according to the ratios outlined in Table 2. The exact division was created by manually examining each scene and selecting samples with representative class-distributions for the test and validation sets. However, due to the limited number of scenes and the significant class imbalance, it was not possible to evenly split the Dronten data into all three sets. As a result, Dronten was excluded from the validation set.

We designed the partitioning to follow a linear increase in scene-count, ranging from 20 to 100 scenes. For this process, we randomly sampled without replacement from the full train set. To determine the number of partitions for a given scene-count, we applied the formula $2 \cdot 100/scene\_count$. This approach ensures a linear relationship between scene-count and the number of partitions with that scene-count, mitigating the impact of smaller sample sizes on dataset variability. Table 3 outlines the resulting partitions.

3.3 Model Training and Tuning

Initially, we attempted to train models using the PointNet++ [14] architecture. However, despite numerous attempts, we were unable to surpass an mIoU of approximately 0.2 on the validation set. Given the unsatisfactory performance, we pivoted to the SPVConv [16] architecture. This change yielded immediate improvements in results, and consequently, we selected SPVConv as the NN architecture for our experiments. Moreover, training new models with SPVConv proved significantly faster than with the PointNet++ model.

We based the initial hyperparameters of the network on SPVConv's results from the SemanticKITTI [1] dataset, subsequently fine-tuning them on our validation set. The parameters can be found in Table 4. Class weights were determined by each class's point distribution within the entire dataset, resulting in the weights $(0.020, 1.25, 20, 16, 23)$ for the respective classes. We chose a batch size of 10, providing a divisor across all partition sizes. The learning rate employed in the final model follows a cosine annealing pattern, starting at 0.2 and ending at 0.001.

We then utilized these same parameters to train models across all partitions, yielding 38 unique models. Through our experimentation, we discovered that smaller partition sizes required more epochs for effective learning from their data. This makes sense due to usage of the same learning rate for all parition sizes. To accommodate this, we scaled the number of epochs linearly based on the size of a partition, applying the formula $3000/scene\_count$.

3.4 Model testing

We conducted standardized testing on each of the 38 models by measuring the class IoU's across two datasets: The test set as outlined in Table 2, and the training partition. Consequently, this testing process yielded $38 \cdot 2 \cdot 5 = 380$ data points, with mIoU calculated as the mean of the five class IoU's.

Parameter	Value
Learning rate	Cosine-annealing: $0.2 \rightarrow 0.001$
Weights	$(0.020, 1.25, 20, 16, 23)$
Epochs	$epochs(size) \coloneqq 3000/size$
Batch size	10
Voxel-size (m)	$x = 0.2, y = 0.2, z = 1$
Range (m)	$x = (-50, 50), y = (-50, 50), z = (-1, 10)$
Input channels	$(x, y, z, intensity)$
Encoder channels	$(32, 64, 128, 256)$
Decoder channels	$(256, 128, 96, 96)$

Table 4. SPVConv model parameters

4 RESULTS

This chapter presents and analyzes the results obtained using the methodology from Section 3. We start by developing best-fit lines for each class, contrasting the IoU when tested on the train and test set with their corresponding scene- and point-count in training data. We employ the Shapiro-Wilk test to assess the normality of the distribution surrounding these fits, and calculate the standard deviations related to the linear regressions. After establishing these regressions, we extrapolate them to the point where the upward trend for test data performance intersects with the downward trend for training data performance. This provides rough estimates of the eventual class-IoU, presuming unlimited training data availability.

4.1 Linear regression fits

Figures 2 and 3 both show class-IoU measurements as a function of the amount of training data, where Figure 2 uses scene-count and Figure 3 uses point-count to represent the amount of data. In both instances, we utilized the complete test-set for training the models as outlined in Section 3.4. Applying Ordinary Least Squares (OLS) methodology, we plotted linear regression lines for each class and observed upward trends, indicating improved performance as more data became accessible. Despite the power-law relationship discussed in Section 2.2, we opted for linear regressions as a best-fit due to the limited amount of available data and measurements.

To examine the quality of the linear regression fits, we calculated the standard deviation and the Shapiro-Wik p-value for each class, considering both test and training data. These results can be seen in Tables Tables 5 and 6. Overall, more than half the distributions for the test set show $p < 0.05$, while only two distributions for the train set show this, indicating a stronger normality-distribution for IoU measured on training data. Looking at the standard deviations of these distributions, we see similar values for the same class, irregardless of scene-count vs. point-count or test set vs. train set.

4.2 Intersections of Learning Curves

Figures 4 and 5 show the intersections between the upward trend of test data performance and the downward trend of training data performance for respectively the scene- and point-count. We observe

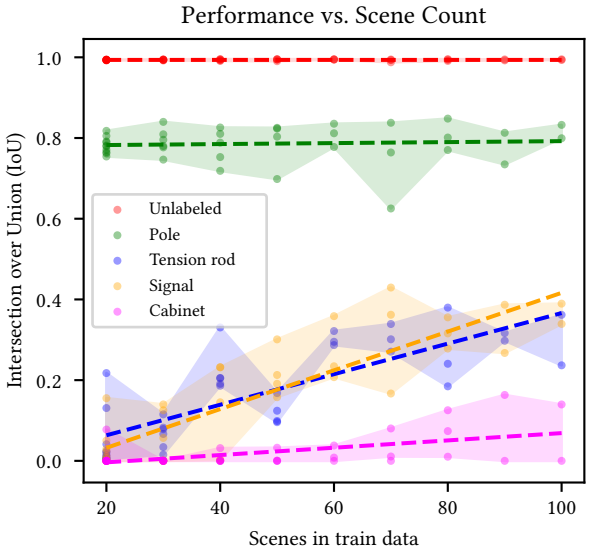

Fig. 2. Performance on test set for various scene-counts in training data

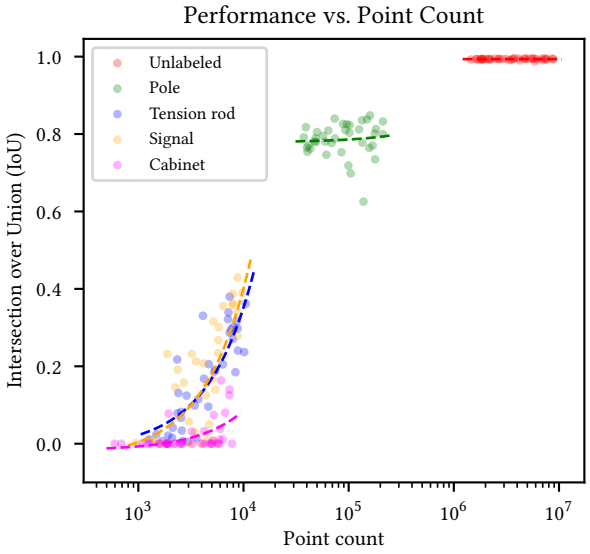

Fig. 3. Performance on test set for various point-counts in training data

considerable differences in the gradient of ascent and descent across the classes; the unlabeled class shows almost no change in IoU, while the gradients for the tension rod are comparatively extremely steep. The charts relating to scene-count and point-count broadly agree regarding the IoU at the intersection points of all classes, with a maximum difference in IoU of 0.11 for the Signal. The mean intersection for point-count shows an IoU of 0.04 higher than the mean intersection for scene-count.

Tables 7 and 8 present the gradients of all linear regressions. The first column represents the ascending gradient on test data, the

Class	Standard deviation		Shapiro-Wik p-val	
	Scenes	Points	Scenes	Points
Unlabeled	1.08E-03	1.08E-03	0.000	0.000
Pole	2.94E-02	2.94E-02	0.001	0.001
Tension rod	3.92E-02	4.50E-02	0.110	0.131
Signal	4.42E-02	5.44E-02	0.199	0.621
Cabinet	2.58E-02	2.64E-02	0.002	0.001

Table 5. Linear regression analysis for test set (grey is better)

Class	Standard deviation		Shapiro-Wik p-val	
	Scenes	Points	Scenes	Points
Unlabeled	1.79E-04	1.80E-04	0.719	0.619
Pole	3.20E-03	3.14E-03	0.116	0.012
Tension rod	3.07E-02	3.74E-02	0.006	0.008
Signal	3.54E-02	3.28E-02	0.143	0.419
Cabinet	2.52E-02	2.82E-02	0.476	0.272

Table 6. Linear regression analysis for train set (grey is better)

Class	Test set	Train set	Ratio
Unlabeled	8.85e-07	-1.87e-05	0.05
Pole	1.56e-04	-4.31e-04	0.36
Tension rod	3.34e-03	-2.28e-03	1.47
Signal	4.43e-03	-5.59e-04	7.92
Cabinet	1.07e-03	-2.29e-03	0.47

Table 7. Linear regression gradients with respect to scene-count

second column represents the descending gradient on training data, and the third column provides the ratio between the two. A ratio greater than one indicates that increases in test performance outpace decreases in training performance. Conversely, a value between zero and one means that training performance decreases more rapidly than test performance increases. A negative value implies that both training and testing performance trends are either both positive or both negative.

In the context of scene-count, a steep upward gradient implies that each additional scene considerably enhances the performance of that class. The steepest upward gradients for scene-count belong to the Tension Rod, Signal, and Cabinet classes; the Pole class is somewhere in the middle, while the Unlabeled class shows very minimal improvement as more scenes are included. Alternatively, when we consider point-count, a sharp upward gradient implies that the performance of that class improves strongly as new points of that class are added to the training set. The findings from point-count are similar to those from scene-count, but more pronounced: the difference in gradients between the Cabinet and Unlabeled classes is approximately 20 times larger.

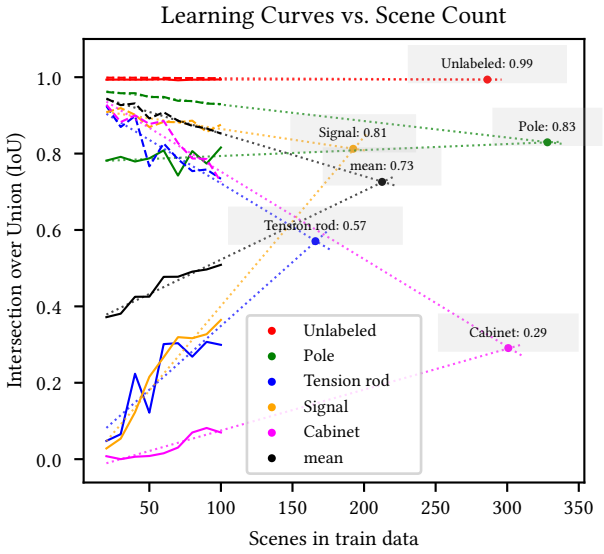

Fig. 4. Performance intersections for various scene-counts in training data

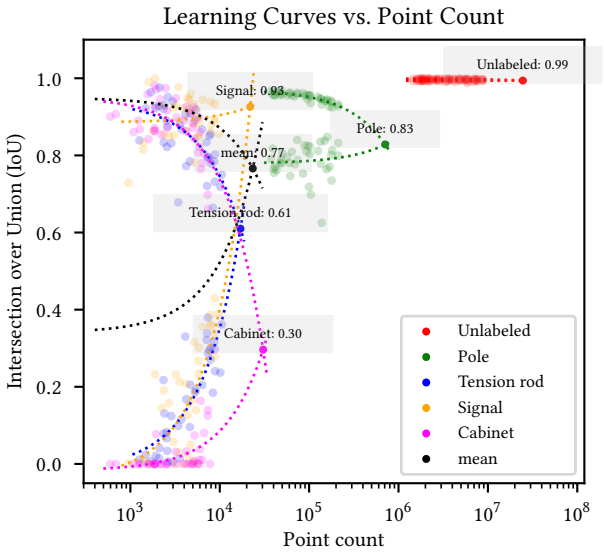

Fig. 5. Performance intersections for various point-counts in training data

Class	Test-data	Train-data	Ratio
Unlabeled	2.30e-11	-2.07e-10	0.11
Pole	6.97e-08	-2.01e-07	0.35
Tension rod	3.66e-05	-1.94e-05	1.89
Signal	4.41e-05	1.85e-06	-23.78
Cabinet	1.03e-05	-2.14e-05	0.48

Table 8. Linear regression gradients with respect to point-count

5 DISCUSSION

In this discussion, we will take a closer look at the results obtained from our experiments, as detailed in Section 4. We start by analyzing the predicted IoU's for each class, focusing on variations related to point-count and scene-count. The purpose of this step is to show the differences in IoU per class and to identify potential methodological issues that may influence the significance of the predictions. Afterwards, we consider the class imbalance as a possible cause for the observed differences in IoU. Lastly, we aim to minimize the impact of this class imbalance on the results, thus allowing us to identify intrinsic difficulties associated to the classes.

5.1 Predicted IoU's

The intersection of learning-curves in Figures 4 and 5 provides estimates for per-class IoU when given infinite training-data. These estimates follow the approach outlined in Section 2.2, fitting linear lines instead of logarithmic lines due to a lack of data. The linear regressions fit relatively well, however (I) not all regressions provide enough evidence for normality around it and (II) the regressions do not extrapolate into infinity. As Hestness et al. outline [6], when the sample-size approaches 0, or as the learning-curves approach another, linear approximations becomes worse. The intersection is therefore indicative of a broad estimate for final IoU given infinite training-data of similar variety and quality present in the current dataset.

What can be concluded from the current dataset is that there is a clear difference in IoU between the classes. This difference is present in the current dataset and remains present when extrapolated to more training-data. Looking at Figure 4 and Table 5, as the the scene-count increases, classes with lower IoU's (Tension Rod, Signal and Cabinet) benefit more than those with higher IoU's (Pole, Unlabeled). This indicates that inter-class variations in IoU will lessen as the sample-size increases, but never dissapearing entirely.

The Signal class is the one outlier regarding the regression gradients, showing ratios for the Signal of 7.92 for scene-count and -23.78 for point-count. These extreme values are due to a very low downward, or even slight upward, gradient when tested on the train partition. The Signal also has high p-values for the Shapiro-Wilk test of normality, indicating skew and kurtosis. It is our expectation that with more data and a better fit, the Signal class would follow the trends of the other classes with a (stronger) downward performance gradient on training data.

5.2 Class imbalance

The significant class-imbalance, as illustrated in Table 1, is a possible cause for the inter-class variations in IoU. From Figure 3 we conclude a positive correlation betwen point-count and IoU, indicating that a higher class-occurence is cause for a higher IoU: The Tension Rod, Signal and Cabinet classes all have a low point-count and comparable IoU's, the Unlabeled class has a high point-count and high IoU and the Pole class is somewhere in the middle for both. An asymptotic relation is apparent, however data is too sparse to properly plot such a line here.

The difference in IoU seems therefore to be caused, at least partially, by the big class-imbalance. This reflects literature on the topic, stating that a significant class-imbalance is often cause for difficulties with learning minority classes [8, 9]. In this case, it appears that even given infinite training-data, the class-imbalance would still cause inter-class variations in IoU.

5.3 Intrinsic difficulty

There are other aspects that may cause difficulties with learning, in addition to the class-imbalance. However, isolating these aspects is difficult due to the influence of the imbalance on learning behaviour of the various classes. When the classes are way out of proportion with respect to point-count, the model has been able to analyze many more features of the majority classes than of the minority classes. Figure 3 shows the difficulty of isolation by comparing performance against point-count instead of scene-count.

A simple way to compare classes without the influence of class-imbalance, is by only comparing those classes with a similar amount of points in the training data. In our case, this allows us to compare the Tension Rod, Signal and Cabinet with another, each having between 0.08% and 0.15% of the data. Between these three classes, the Tension Rod and Signal have very similar learning-gradients. The Cabinet, however, has a gradient around 4 times shallower, meaning that the model has much more difficulty to classify Cabinets correctly than Tension Rods or Signals when trained on the same amount of data. This is also visible in the absolute IoU's of each class when trained on $5 \cdot 10^2$ points, resulting in an absolute difference of about 4 times.

One can also look at the downward gradient of performance on the train partition; the steepness of this gradient tells us about the maximum achievable IoU instead of the learning-rate. The linear-regression fits for these samples may not be accurate enough to provide strong conclusions, with Table 6 showing high Shapiro-Wik p-values to prove normal-distribution. What is interesting is that, while the Tension Rod and Signal have a steeper upward gradients than the Cabinet, the downward gradients of the Tension Rod and Cabinet are actually very similar, while the gradient of the Signal is much shallower. This indicates that the model actually overfits less for the Signal, therefore predicting a higher final IoU.

6 CONCLUSION

Given the high cost associated with the creation of big datasets, it is paramount to be able to analyze the learning behavior of NNs on smaller datasets. This circumvents the immediate need of creating a comprehensive dataset, while still allowing us to make predictions about the performance on a comprehensive dataset. Through an analysis of the learning behaviour of the SPVConv NN on a small railway dataset, we analyze semantic segmentation performance with respect to point- and scene-count. This allows us to extrapolate class-IoU's to bigger datasets and analyze the causes of identified differences in IoU between the classes.

Our findings suggest that the variations in IoU between the classes is partly caused by the big class-imbalance in the dataset; this appears to be an asymptotic relation between point-count and IoU. This relation does not just hold for the limited dataset, but also appears to hold as the amount of available data increases. In addition to the observed class-imbalance, we identify additional factors

impacting learning rates and IoU: With an equal amount of training data, the Cabinet class is comparatively much harder to learn than both the Signal and Tension Rod classes. In contrast, overfitting presents less of a problem for the Signal class, as evidenced by the shallower downward gradients on training data compared to the Tension Rod and Cabinet classes.

Though this research offers a novel approach and presents some interesting findings related to learning behaviour, the predictive power of the learning curves could be improved dramatically with increased amounts of data: more models trained on the current dataset and on bigger datasets would be very useful for this analysis. Validating the findings in this research through measurements on big, publically available datasets like SemanticKITTI is a good idea. It would also also be worth applying our methodology to various state-of-the-art NN-architectures and change network parameters like encoder/decoder channels or voxel-size. PointNet++ was initially tried out for our analysis but did not deliver good results; other models that test better on the SemanticKITTI dataset could provide valueable insights. An additional area of interest is to identify object-count instead of point-count as a measure of data availability, which may show clearer correlations to IoU.

REFERENCES

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. 2019. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*.

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. arXiv:1904.08755 [cs.CV]

[3] Teodor Fredriksson, David Issa Mattos, Jan Bosch, and Helena Holmström Olsson. 2020. Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies. In *Product-Focused Software Process Improvement (Lecture Notes in Computer Science)*, Maurizio Morisio, Marco Torchiano, and Andreas Jedlitschka (Eds.). Springer International Publishing, Cham, 202–216. https://doi.org/10.1007/978-3-030-64148-1_13

[4] Eleonora Grilli, Fabio Menna, and Fabio Remondino. 2017. A Review of Point Clouds Segmentation and Classification Algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42 (2017), 339.

[5] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. 2020. Scaling Laws for Autoregressive Generative Modeling. arXiv:2010.14701 [cs.LG]

[6] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. 2017. Deep Learning Scaling Is Predictable, Empirically. https://doi.org/10.48550/arXiv.1712.00409 arXiv:1712.00409 [cs, stat]

[7] Marcus Hutter. 2021. Learning Curve Theory. https://arxiv.org/abs/2102.04074v1.

[8] Nathalie Japkowicz and Shaju Stephen. 2002. The Class Imbalance Problem: A Systematic Study. *Intelligent Data Analysis* 6, 5 (Jan. 2002), 429–449. https://doi.org/10.3233/IDA-2002-6504

[9] Justin M. Johnson and Taghi M. Khoshgoftaar. 2019. Survey on Deep Learning with Class Imbalance. *Journal of Big Data* 6, 1 (March 2019), 27. https://doi.org/10.1186/s40537-019-0192-5

[10] Felix Mohr and Jan N. Rijn. 2022. Learning Curves for Decision Making in Supervised Machine Learning: A Survey. arXiv:2201.12150 [cs.LG]

[11] Kyuetaek Oh, Mintaek Yoo, Nayoung Jin, Jisu Ko, Jeonguk Seo, Hyojin Joo, and Minsam Ko. 2022. A Review of Deep Learning Applications for Railway Safety. *Applied Sciences* 12, 20 (Jan. 2022), 10572. https://doi.org/10.3390/app122010572

[12] Claudia Perlich. 2010. Learning Curves in Machine Learning.

[13] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. arXiv:1612.00593 [cs.CV]

[14] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. arXiv:1706.02413 [cs.CV]

[15] Jessada Sresakoolchai and Sakdirat Kaewunruen. 2021. Detection and Severity Evaluation of Combined Rail Defects Using Deep Learning. *Vibration* 4, 2 (June 2021), 341–356. https://doi.org/10.3390/vibration4020022

[16] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. 2020. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *Computer Vision – ECCV 2020 (Lecture Notes in Computer Science)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 685–702. https://doi.org/10.1007/978-3-030-58604-1_41

[17] Bram Ton, Faizan Ahmed, and Jeroen Linssen. 2023. Semantic Segmentation of Terrestrial Laser Scans of Railway Catenary Arches: A Use Case Perspective. *Sensors* 23, 1 (Jan. 2023), 222. https://doi.org/10.3390/s23010222

[18] Chunsheng Yang, Yanmin Sun, Chris Ladubec, and Yan Liu. 2021. Developing Machine Learning-Based Models for Railway Inspection. *Applied Sciences* 11, 1 (Jan. 2021), 13. https://doi.org/10.3390/app11010013

[19] Dimitris Zermas, Izzat Izzat, and Nikolaos Papanikolopoulos. 2017. Fast Segmentation of 3D Point Clouds: A Paradigm on LiDAR Data for Autonomous Vehicle Applications. https://doi.org/10.1109/ICRA.2017.7989591

[20] Jiaying Zhang, Xiaoli Zhao, Zheng Chen, and Zhejun Lu. 2019. A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. *IEEE access : practical innovations, open solutions* 7 (2019), 179118–179133. https://doi.org/10.1109/ACCESS.2019.2958671