Check for updates

OPEN ACCESS

EDITED BY Salman Azhar, Auburn University, United States

REVIEWED BY

Rizwan Farooqui, Mississippi State University, United States Rana Muhammad Irfan Anwar, Auburn University, United States

*CORRESPONDENCE David Crampen, crampen@gia.rwth-aachen.de

RECEIVED 07 April 2025 ACCEPTED 22 May 2025 PUBLISHED 11 June 2025

CITATION

Crampen D and Blankenbach J (2025) Data enrichment for semantic segmentation of point clouds for the generation of geometric-semantic road models. *Front. Built Environ.* 11:1607375. doi: 10.3389/fbuil.2025.1607375

COPYRIGHT

© 2025 Crampen and Blankenbach. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Data enrichment for semantic segmentation of point clouds for the generation of geometric-semantic road models

David Crampen* and Joerg Blankenbach

Civil Engineering Department, Geodetic Institute and Chair for Computing in Civil Engineering and GIS, RWTH Aachen University, Aachen, Germany

Digitalizing highway infrastructure is gaining interest in Germany and other countries due to the need for greater efficiency and sustainability. The maintenance of the built infrastructure accounts for nearly 30% of greenhouse gas emissions in Germany. To address this, Digital Twins are emerging as tools to optimize road systems. A Digital Twin of a built asset relies on a geometric-semantic as-is model of the area of interest, where an essential step for automated model generation is the semantic segmentation of reality capture data. While most approaches handle data without considering realworld context, our approach leverages existing geospatial data to enrich the data foundation through an adaptive feature extraction workflow. This workflow is adaptable to various model architectures, from deep learning methods like PointNet++ and PointNeXt to traditional machine learning models such as Random Forest. Our four-step workflow significantly boosts performance, improving overall accuracy by 20% and unweighted mean Intersection over Union (mIoU) by up to 43.47%. The target application is the semantic segmentation of point clouds in road environments. Additionally, the proposed modular workflow can be easily customized to fit diverse data sources and enhance semantic segmentation performance in a model-agnostic way.

KEYWORDS

point clouds, semantic segmentation, deep learning, machine learning, geometricsemantic road models, automation, digital twin, data enrichment

1 Introduction

The implementation of Digital Twins of roads has great potential in increasing the efficiency in maintenance and management of dense road infrastructure networks. A Digital Twin leverages multi-modal data to support and, in the future automize decision-making (Federal Ministry for Digital and Transport, BMDV Germany, 2021). Digital twins of roads can be extremely useful for many applications, such as predictive maintenance, traffic control or autonomous driving. However, for establishing a Digital Twin use case in the road environment, a digital geometric-semantic representation is required to act as a foundation for georeferencing sensors and relevant components in the environment and provide a container to store and interlink relevant data in the form of an as-is model (Noroozinejad Farsangi et al., 2024). To create such a representation, reality-capturing technologies are indispensable. The process from data acquisition to the final



digital as-is model is composed of several steps. Performing those manually is very time-consuming and linked to major modeling effort and costs. Therefore, automating this process has enormous time- and cost-saving potential. In Building Information Modeling (BIM), regardless whether automated or performed manually, the Scan-to-BIM process targets the built environment without an existing, up-to-date digital representation. While BIM aims at the aggregation of all relevant information about the asset and acts as an information container for all available data, this is not necessarily desired for a Digital Twin use case (Crampen et al., 2024). Irrelevant information is a burden for simulation and synchronization between the physical world and the virtual world. Therefore, the digital representation should only inherit the information necessary for the use case conducted. In that way, a use case defines requirements for the underlying digital representation. The resulting Scan-to-Twin workflow is schematically illustrated in Figure 1.

The workflow consists of a sequential chain of data acquisition, data processing, geometry extraction, and model generation, with the data processing step being our primary focus. This step is crucial as it introduces semantic information into the data collection, directly impacting the quality and robustness of the derived model. Since the accuracy of semantic object class segmentation greatly influences the detail and reliability of the resulting model, we leverage prior knowledge from geospatial data to enrich the data foundation and enhance the performance of various data-driven segmentation methods. Additionally, our workflow is designed with modularity in mind, making it adaptable to different data sources and supporting both georeferenced and unreferenced point cloud data. There are various techniques employed in the data processing step, which are introduced in Section 2.

The rest of this contribution is structured as follows: Section 2 reviews the state of research; Section 3 summarizes the identified research gaps and goals of our contribution; and Section 4 describes our workflow, which includes a uniform pre-segmentation

step followed by an automatic feature extraction process that automatically adapts to the dataset. Section 5 presents our results. The impact of our preprocessing workflow is evaluated using different approaches by comparing model performance at each stage of the preprocessing scheme to highlight performance enhancements achieved in each consecutive step. In Section 6, we discuss our findings and possible future directions, particularly in linking our workflow to model generation for Digital Twins of roads, while we summarize our work in Section 7. The results of an ablation study conducted in the course of our analyses and visual examples of the final model predictions can be reviewed in the Supplementary Material.

2 State-of-research and related work

One major challenge for automation of the Scan-to-Twin process is its sequential nature, which leads to multiple interfaces and error propagation. This can lead to large errors in the output of the workflow, especially if inaccuracies occur in early steps (Esfahani et al., 2021). Consequently, relying on uncertain semantic objects may introduce intractable errors in the final geometricsemantic model (GSM) (Vassilev et al., 2024). Although there are several approaches to directly extract structures' geometries for indoor purposes, such as Martens and Blankenbach (2023), such approaches are difficult to apply in outdoor environments due to many natural objects and the large spatial extent, leading to extremely sparse voxel grids. Extracting the relevant objects through semantic segmentation in a first step, however, leads to a better foundation for geometry reconstruction in an outdoor environment (Grandio et al., 2022). Continuously, new deep learning architectures introduce new ways of capturing significant aspects of the point cloud data. Still, many works indicate, that most deep learning models meet a trade-off between local and

global context. This trade-off has been recognized since the first architecture working directly on point cloud data was published in Qi et al. (2017a). Since then, the complexity of new architectures has largely increased. Qian et al. (2022) indicate, that using better preprocessing techniques, the performance of less recent deep learning architectures can potentially exceed the performance of the latest and way more complex architectures. Introducing geospatial data (often also referred to as geodata or GIS data) to perform presegmentation as a preprocessing step for semantic segmentation could prove beneficial for the integration of prior knowledge and improve the perfomance of arbitrary models by increasing the data informativeness instead of the complexity of the model. While the available computational power has increased rapidly over the last decade, making complex deep learning architectures applicable, earlier methods did not rely on data, but on a model, formulating rules as well as mathematical filters based on heuristics and other methods to dissect data into characteristic segments. Today, methods for point cloud segmentation can be categorized into two groups: Point cloud segmentation (PCS) and point cloud semantic segmentation (PCSS). While the categorization might indicate that research is progressing from PCS to PCSS, there are several aspects currently limiting the broad application of PCSS for geometric-semantic model reconstruction, with a major factor being data availability. Therefore, significant work is conducted to develop hybrid approaches, as in Schatz and Domer (2024), to mitigate the respective issues arising in both sets of methods.

2.1 Point cloud segmentation (PCS)

In PCS, filters, rules, and static algorithms are commonly used to segment point clouds. Early approaches, were edge detection applied to range images (Bhanu et al., 1986) or region growing for surface approximation (Besl and Jain, 1988). Today similar methods are utilized for preprocessing, such as feature extraction, as demonstrated in recent studies (Grilli et al., 2017). Justo et al. (2021) used region growing with voxelized mobile laser scan (MLS) data to extract the centerlines of road segments in order to derive a digital model. Yan and Hajjar (2021) applied a region-growing algorithm to extract bridge girders from point clouds.

Various techniques, such as Hough Transform and RANSAC (Random Sample Consensus), are extensively used in fitting techniques to fit diverse shapes into point cloud data, enabling the extraction of objects with characteristic shapes. Safaie et al. (2021) utilized Hough Transform to extract tree stems near roads by vertically slicing the point cloud data and fitting circles to the vertical cross-sections of the stems. Li and Vosselman (2018) applied Hough Transform to extract pole-like objects, including road signs, traffic lights, and road lamps, from Mobile Mapping System (MMS) data. Wang et al. (2022) improved the Universal-RANSAC algorithm to perform ground filtering on MMS point clouds, simplifying the semantic segmentation of objects by robustly removing ground points.

Edge features, widely used in image segmentation, play a significant role in direct object extraction. The detection of changes in local curvature extends beyond images and applies to point clouds, as well as rough changes in normal orientation, gradients, intensity, or principal curvature within point clouds to indicate object transitions. In that way, Rato and Santos (2021) utilized edge detection in MMS point clouds for vehicle positioning in autonomous driving, while Wang et al. (2019) incorporated edge features in their artificial neural network architecture to enhance the ability of distinguishing different objects by constructing local neighborhood graphs. For extracting pole-like objects from MMS data, Li Y. et al. (2019) first applied elevation filtering and combined region growing with multiple filtering techniques and unsupervised learning to extract different object classes from an urban road environment, while Lucas et al. (2019), Soilán et al. (2017) and Ma et al. (2017) used reflectance-based features to filter vegetation road markings, respectively, leveraging information provided by most laser scanners.

2.2 Point cloud semantic segmentation (PCSS)

PCSS uses machine learning methods to classify point cloud segments into semantic classes, that involves supervised machine learning, semi-supervised as well as unsupervised machine learning approaches. Supervised machine learning can be further differentiated into conventional machine learning, where manual feature engineering has to be preliminary performed to fit a model to the feature set, and deep learning, where artificial neural networks learn abstract features themselves to allow a classification of different semantic classes. A classical machine learning model used for PCSS are Random Forests, which have been used for instance to segment pole-like structures from MMS data (Wang et al., 2021). Biçici and Zeybek (2021) applied random forest classifiers to airborne laser scan (ALS) data to extract road surfaces.

The unstructured nature of point clouds is taken into account by different deep learning approaches that attempt to structure the data or were designed to deal with the point cloud directly to form connections during training by passing groups of points to the neural network at once. Although methods employing spectral images and voxelization have been early approaches, there are several recent works dealing with outdoor point cloud segmentation, such as Xu et al. (2021). Another approach applied to outdoor point clouds is presented in Roynard et al. (2018a), where a voxel-based neural network was developed, achieving state-of-the-art performance on a terrestrial laser scanning (TLS) outdoor dataset. A popular approach leveraging spectral-images is SnapNet (Boulch et al., 2017), which mitigates the structuring issue by generating images from the meshed point cloud.

As opposed to the previously discussed methods, the deep learning architectures working directly on unstructured point clouds eliminate the problems introduced in the structuring process, retaining the information inherent to the point cloud and requiring less computational effort to form structures (Zhang et al., 2019). The first method in this domain was PointNet (Qi et al., 2017a), where a symmetric function is applied to order the point cloud and enable the use of multi-layer perceptrons to learn point-wise features that are aggregated to obtain global features for segmentation. For road segmentation, PointNet++ (Qi et al., 2017b) is often used. Ma et al. (2022) for example, used PointNet++ for road footprint segmentation, whereas in Shin et al. (2022) it was used for building extraction from ALS data. Recently, transformer-based networks have been adapted to point clouds, using an attention mechanism, which was first introduced in the natural language processing domain, to outperform most other architectures (Zhao et al., 2021). There are many approaches being published every year that regularly set new measures in state-of-the-art performance on well-known benchmarking datasets such as ModelNet (Wu et al., 2015), S3DIS (Armeni et al., 2016) or Paris-Lille-3D (Roynard et al., 2018b) to name a few. Interestingly, in Qian et al. (2022) it is shown with their PointNext architecture that updating the training strategy and adding rather small modifications to the model architecture of PointNet++ can lead to a model outperforming most of the recent approaches, indicating that at some point the leverage of higher model complexity on actual performance becomes smaller than that of an improved training approach and potentially data preprocessing workflow.

2.3 Prior knowledge and feature extraction

Regarding preprocessing, especially encoding prior knowledge into the data directly has large potential for performance enhancement of arbitrary models. Li F. et al. (2019), for example, first split their MMS data into blocks and then removed ground points through local height variance as a first step of their workflow for road furniture extraction. A standard method is separating ground and above ground points to create two separate classification problems (Baligh Jahromi et al., 2011; Chen et al., 2021; Shi et al., 2018; Sithole and Vosselman, 2001; Yilmaz, 2021). In Murtiyoso et al. (2020), roofs were extracted using available geospatial data of building footprints. In Paden et al. (2022) ALS point clouds and geospatial data of building footprints were used to directly create CityJSON data by extruding the footprints by the height of the inlying point cloud segments. In a previous study (Murtiyoso and Grussenmeyer, 2019) geospatial data has been used to extract point cloud segments of heritage complexes and enrich the segments with the information present in the geospatial data. Aljumaily et al. (2019) used Open Street Map (OSM) data in the same way to provide segmented point clouds with richer semantic data. Lastly, in Park and Guldmann (2019) OSM data was used for building extraction using building footprint data to cut point cloud segments out of the point cloud.

Typical features used to enrich point cloud data are eigenvaluebased features (Bremer et al., 2013). Higher-order features can be derived from various transformations or preprocessing techniques that can be called descriptor-based features, like in Alexiou et al. (2024). Others involve spin images, shape distributions (Osada et al., 2002), or histograms (Rusu et al., 2009). In road environments, height gradient features can indicate the height change when reaching the road edge if it is bound by a curb, as done in Yang et al. (2020). Using the timestamps on MMS data, the road orientation can be retrieved. Voxel-based region-growing approaches use seed points as features, like in Hirt et al. (2021). The resulting segments can be used to extract other higher-order features, as done by Zhang et al. (2022).

Another significant aspect of using extracted features is the selection of the most meaningful ones. To do this heuristically, there are different solutions, such as Recursive Feature Elimination (RFE) Schlosser et al. (2020) or filter-based methods Atik and Duran (2022). Tan et al. (2007) developed a Genetic algorithm (GA) that

heuristically finds the best set of features from a set of populations of features and improves the set iteratively through mechanisms of recombination, reproduction, and mutation. Compared to filterbased methods, GA is a model-agnostic wrapper-based approach. Therefore, it is applicable to arbitrary models using diverse fitness functions. In comparison to RFE, which is also wrapper-based, the advantage of GA is the capability of conducting a global search, while RFE works greedy and therefore more efficient, which however may lead to suboptimal feature sets. Road segmentation on point clouds involves various approaches from both PCS and PCSS, primarily utilizing data captured from MMS. However, MMS are designed to capture the near field of a road within the scanner system's field of view, while many use cases require more extensive coverage beyond what an MMS system can capture. Connecting multiple data sources during the capturing stage can leverage synergies and enhance connections between different domains, including urban planning, traffic simulation, and risk management. However, combining different point cloud data sources can be challenging due to heterogeneity in density, features and scan patterns that make it harder for a model to generalize well (Hu et al., 2023). Still, establishing a data sourceagnostic workflow that maximizes the quality of the semantic segmentation is necessary to enable automatic model generation. Especially since models for Digital Twins have to stay up-to-date and repeated manual modeling in relatively short time intervals is simply not feasible (Mansour et al., 2024).

3 Gaps and research questions

The research gaps addressed within this contribution are as follows: (1) Development of a strategy to enrich point cloud data of roads for semantic segmentation. (2) Efficient leveraging of prior knowledge to improve the output of an arbitrary machine learning approach. (3) Generalized data enrichment scheme to automatically adapt to different point cloud data sources. Therefore, this contribution aims to answer the following research questions:

- 1. How can reality capture data of roads be enriched for semantic segmentation purposes without limiting the applicability to a specific data source? (RQ1)
- 2. What prior knowledge can be leveraged to further enhance information richness of point cloud data of roads? (RQ2)
- 3. What methods can be combined to improve the overall fit of point cloud data of roads to machine learning applications without limiting optimization to a specific model? (RQ3)
- 4. What is the best approach to make the workflow automatically adapt to changes in data source and data quality? (RQ4)

This contribution introduces a workflow specifically designed for performing pre-segmentation and preprocessing for Point Cloud Semantic Segmentation (PCSS) applications on largescale laser scanning data. Through our automatic preprocessing workflow, we showcase improvements in various PCSS models, including standard machine learning methods such as bagging and boosting methods for decision trees, as well as deep learning-based techniques like PointNet++ (Qi et al., 2017b) and PointNeXt (Qian et al., 2022). This improvement is achieved while significantly reducing the required amount of data and training time. By combining PCS and PCSS with prior knowledge, we leverage the abundant ALS point cloud data provided by governments as open data. Our approach combines automatic adaptation to input data with a modular structure, enabling the flexible application of our process to different point cloud related tasks. This distinguishes our method form other task specific solutions to point cloud semantic segmentation approaches which are often tailored to specific datasets. With this strategy, we aim at ensuring ease of use and a high degree of generalizability for different point cloud related tasks. We validate this claim by comparing the model performances on ALS data as well as MMS data.

4 Methodology

Our workflow consists of four steps, which can be executed sequentially or individually, based on the characteristics of the available data (Figure 2):

- 1. Prior knowledge in the form of existing geospatial data is utilized to perform pre-segmentation of roads from ALS data by generating horizontal boundaries for the area of interest.
- 2. Existing geospatial data is again used to identify road blocks along the trajectory of the roads. These blocks are then transformed to a uniform orientation, which improves the regularity of the input for feature extraction.
- 3. Various feature extraction methods are employed to generate an initial set of features.
- 4. A feature selection method based on a custom GA is applied to reduce the initial feature space to a more condensed set of features. This selection process helps to improve the efficiency and effectiveness of the dataset.

The workflow is structured into two phases: pre-segmentation and feature generation (Figure 2). This design allows evaluation of performance enhancements achieved by applying each phase independently. It also enables the use of either phase separately to accommodate data that is not georeferenced and, therefore, not suitable for Phase 1. While our workflow combines different approaches and adaptations from other domains, we want to emphasize that the focus of this contribution is not merely investigating the performance of deep learning models. However, we test the preprocessing scheme outputs on various models from deep learning and machine learning as a proof of the versatility and universal impact of our preprocessing. Specifically, we use PointNet++ and the more recent PointNeXt as proxies to show the impact of our data enrichment process for deep learning approaches, while training decision tree-based models such as Random Forest and simple multi-layer perceptrons (MLP) as a comparison.

4.1 Dataset

The data used for our workflow is open data accessible from "geoportal.nrw" (Interministerial Committee, 2025). This platform provides ALS point clouds and further geospatial data specifically for the North Rhine-Westphalia region in Germany. The ALS point cloud data has an average density of approximately 20 points per square meter. It contains several scanner features, of

which intensity and number of returns are most relevant for road segmentation. The number of returns is captured by partial reflections of the same beam that are registered multiple times, indicating a transparent or semi-transparent surface, while the intensity is the strength of the reflection received, which differs for surfaces with different reflectivity characteristics. Both the ALS point cloud data and the geospatial data are georeferenced in the official ETRS89/UTM32 coordinate reference system of Germany, ensuring spatial alignment between the datasets. Since the ALS point clouds did not include RGB data initially, we employ a simple spatial interpolation technique using a photogrammetric point cloud of the same region to assign color values to it. In this work, we focused on extracting primary and secondary roads, including highways and federal roads. This selection criterion helped to narrow down the scope of the segmentation task. To demonstrate the effectiveness of Step 1, the pre-segmentation process, we observed a substantial reduction in the number of points in each segment. On average, applying Step 1 reduced the amount of data by 95%, narrowing down the scope to the area of interest. Two examples of the twelve used tiles are shown in Figure 3.

The MMS dataset used was captured using a vehicle-mounted mapping system Trimble MX9 by the german road agency Straßen.NRW and comprises small scenes of highways close to bridges and shield gantries in the State of North-Rhine-Westafalia. The average density of this dataset is approximately 700 points per square meter. It consists of localized coordinates and intensity values, while lacking color information. To create the dataset, we manually labeled 37 of those section sections, where 32 were used for training 2 for validation and 3 for testing. For the annotation of the semantic object classes of both datasets, the open source software CloudCompare was used.

4.2 Phase 1: pre-segmentation

For coarse pre-segmentation of the road space from the ALS data, we apply an approach comparable to the idea presented in Murtiyoso et al. (2020), but apply existing geospatial data of the road network to extract the road environment as a region of interest, instead of cutting using building footprints to cut out buildings. While different file formats such as GML, XML or ESRI Shapefile are applicable, we tested our approach on OSM data and German ATKIS (the Official Topographic-Cartographic Information System) data, which is generally more precise than OSM data. One advantage of using the above-mentioned geospatial data is its nationwide availability. The Shapefile from OSM and ATKIS consist of simple polylines and semantic information, which is attached to each polyline. We use the functional classification in the data to determine respective maximum road widths for different road classes in order to create polygons from the polylines by buffering, and then utilize these to automatically slice out our area of interest.

4.2.1 Step 1: cutting point clouds with geospatial data

The data of the whole road network of NRW is cut automatically by using the boundaries of the point clouds to first filter the geospatial data and then apply the buffered polygons to the point



clouds, which are provided in tiles of 1 square kilometer. This is schematically shown in Figure 4 and largely reduces search time for matching geospatial data and point clouds.

The point clouds are cut using a path-cutting algorithm. After the horizontal pre-segmentation, we utilize a cloth simulation filter (CSF) to separate ground and above ground points. The ground and above-ground points are then separately treated for the rest of the workflow. For ground points, the classes for segmentation are road, non-road ground and road markings, while the above ground points are segmented into road furniture, vehicles and vegetation.

4.2.2 Step 2: road block generation

One advantage of using road-related geospatial data is the possibility to determine the road orientation, which can be used to process data quite similar to the trajectory information obtained using MMS data timestamps. We sample an arbitrary number of points on the polyline and use the predecessor-successor relation of these points to construct rectangular blocks along the road segment with sample point distance and road width as input parameters. This allows us to further extract smaller road segments and rotate them uniformly around the height axis to obtain homogenous road blocks. This procedure enables the extraction of features along the road trajectory. Figure 5 schematically shows the procedure of Step 2.

The road blocks are used in the second Phase 2 to extract additional histogram-based features. After the features have been extracted from the individual blocks, the point cloud is reconstructed by rotating the road block back to their original positions. When using the road block approach, we add the localized Northing and Easting UTM coordinates as features and create bins of points with uniform thickness along the principal axis as a basis for the histograms. In each bin, we compute the mean intensity and the intensity gradient over the bins as additional features. When visualizing the intensity gradients of the ground points along the road width, a characteristic pattern can be recognized, indicating a good potential for separation of lanes and road markings in one direction. Figure 6 shows three examples of histograms of the intensity gradients and their corresponding road segments.









4.3 Phase 2: feature generation

In the second phase, we extract features from the point cloud. Preliminary analyses showed that extracting all features directly from road blocks introduces unnecessary edges to the receptive fields, and thereby results in less informative features than if applying the feature extraction to the whole point cloud. We first extract features and optimize them by varying the parameters of the applied approaches for extraction, which is validated using a simple multilayer perceptron classifier (MLP) that is trained with ten manually labeled road segments as a training set and measuring the accuracy reached with the given features on a validation set consisting of two road segments. After this, we employ a custom GA for feature selection, thereby reducing the initially large feature set of up to 86 features depending on the used steps to a more condensed set of features. The choice of GA results from our aim towards the model agnostic design of our workflow. We validate the feature selection step by training an MLP on the new set of features using the same training split as in the previous step to verify whether model performance can be maintained using fewer features.

4.3.1 Step 3: feature extraction

The extracted features are grouped into: eigenvalue-based features, reflectance-based features, height-based features, that are computed by applying a cylindrical neighborhood search and computing the local distribution of height and other basic

TABLE 1 Initial feature set grouped by extraction module, and area of application.

Feature	Extraction module	Application area					
XYZ (mult.)	Position	Global					
RGB (mult.)	Colors	Global					
Number of Returns	Reflectance	Global					
Intensity	Reflectance	Global					
Planarity	Eigenvalue	Local Neigh.					
Linearity	Eigenvalue	Local Neigh.					
Sphericity	Eigenvalue	Local Neigh.					
Verticality	Eigenvalue	Local Neigh.					
Mean Height	Local Distribution	Local Neigh.					
Height Standard Dev.	Local Distribution	Local Neigh.					
Mean Intensity	Local Distribution	Local Neigh.					
Intensity Standard Dev.	Local Distribution	Local Neigh.					
Height Range	Local Distribution	Local Neigh.					
Number of Neighbors	Local Distribution	Local Neigh.					
Eigenvalue Ranges (mult.)	Local Distribution	Local Neigh.					
Height Difference to Ground	Ground Relational	Local Neigh.					
Custom Height Features (mult.)	Ground Relational	Local Neigh.					
DB Mean Height	DBSCAN	Segment					
DB Standard Dev. Height	DBSCAN	Segment					
DB Mean Intensity	DBSCAN	Segment					
DB Standard Dev. Intensity	DBSCAN	Segment					
DB Custom Shape features (mult.)	DBSCAN	Segment					
Eigenvalue Segment features (mult.)	DBSCAN	Segment					
Ransac Plane	RANSAC	Segment					
Ransac Cube	RANSAC	Segment					
Ransac Sphere	RANSAC	Segment					
Ransac Cylinder	RANSAC	Segment					
Ransac Line	RANSAC	Segment					

(Continued on the following page)

TABLE 1 (Continued) Initial feature set grouped by extraction module, and area of application.

Feature	Extraction module	Application area				
Ransac Circle	RANSAC	Segment				
Intensity Gradients X and Y (mult.)	Road Blocks	Bins along X and Y				
Mean Intensity	Road Blocks	Bins along X and Y				
Intensity Gradients X	Road Blocks	Bins along X				
Intensity Gradients Y	Road Blocks	Bins along Y				
Gaussian Mixture model	GMM	Point-wise				
K-Means	K-Means	Point-wise				

features, segment-based features, which are obtained by using unsupervised approaches such as generalized Random Sample Consensus (RANSAC) for model fitting as well as Density-Based Spatial Clustering of Application with Noise (DBSCAN). We apply shape constraints to the resulting segments to obtain segment-based shape descriptor features. For the segment features computed with RANSAC, we simply encode the recognized shapes, which are: cube, plane, cylinder, circle, line and sphere, as binary features, for belonging or not belonging to the inliers of a fit shape. Lastly, we use a Gaussian Mixture Model and k-Means clustering to cluster points into six clusters based on the previously extracted features and add the cluster labels to the point-wise features. Table 1 shows the full list of extracted features.

All extracted features either directly or indirectly depend on the receptive field that is steered by the neighborhood radius for neighborhood-based features and the maximum distance in the case of DBSCAN-based features. Therefore, we conducted an ablation study by varying the parameters controlling the receptive field for feature extraction. We compute the features in six separate groups to find a trade-off between computational expense and flexibility in feature optimization. Optimizing the reception radius for each single feature is the upper boundary, and optimizing one uniform radius for all features is the lower boundary of the tradeoff. The resulting groups are named as follows: Eigenvalue-features, LocalDist-features, DBSCAN-features, RANSAC-features, Ground-Relation-features and RoadBlock-features, where the second last group is obviously only applicable to the above ground points after ground filtering the results of the ablation study conducted are detailed in the Supplementary Material Section. For the RoadBlockfeatures, we create bins with a fixed width along the direction of travel, which is denoted with Y, as well as orthogonal to the direction of travel, denoted with X to assign points inside each bin to intensity gradients over consecutive bins. A schematic visualization of this procedure is shown in Figure 7.

Since varying the radius to obtain the best features on a full dataset is time consuming, we also compare the manual reception radius optimization with the use of the optimal neighborhood



definition, which can be used to automatically find optimal k for k-nearest neighborhood search per-point. We compute the optimal neighborhood according to Weinmann et al. (2014) as shown in the equations below Equations 1–3:

$$E_{\lambda} = -e_1 ln(e_1) - e_2 ln(e_2) - e_3 ln(e_3)$$
(1)

$$e_i = \frac{\lambda_i}{\sum \lambda}, \ i \in \{1, 2, 3\}$$
(2)

$$\lambda_1 \ge \lambda_2 \ge \lambda_3 \ge 0 \tag{3}$$

However, instead of using k-nearest neighborhood, we use radius neighborhood in a range between 1 m and 7 m, with step sizes of 0.5 m, to being able to pass the identified reception radius to other feature extraction modules for improved adaptiveness. The segmentbased features using DBSCAN rely on a criterion that depends on point cloud density. Therefore, we choose Epsilon according to a fraction of the average number of neighbors computed in the optimal neighborhood approach. To increase variability of the receptive field, we also used the k-nearest neighborhood definition for additional feature extraction, with an inverse relation compared to radius neighborhood, increasing k for small radius neighborhoods.

4.3.2 Step 4: feature selection

For feature selection, we utilize a custom GA that iteratively searches for the best feature set. We additionally compare the performance of the reduced feature set to the performance of the initial feature set to test the impact of reducing the feature set by employing Step 4. Our custom GA, is configured to run 10 epochs, aggregating a population of 20 feature sets with a specified number of features and using 3-fold cross-validation of a Naïve Bayesian classifier (Huang and Li, 2011) on random subsamples of 100,000 points with a matching class distribution of the full dataset for each fold. The results are ranked according to their unweighted mIoU and the first five ranking feature sets are transferred to the next epoch, while ten new feature sets are generated by randomly combining the five best feature sets from the previous epoch. The number of recombining sets is reduced in each epoch to make the optimization converge. We additionally allow random mutation with a probability of 5% in all features of a new set, limiting the maximum number of mutations to 10% of the feature count, while choosing the remaining 5 sets from again randomly sampling features from the full feature set. After each epoch the best feature set is validated on the full validation set to ensure that the result is representative.

4.4 Evaluation strategy

To evaluate the impact of the stages of our workflow, we compare different configurations of our workflow. The resulting six configurations are depicted in Figure 8. The colored markers highlight the most relevant stages to clarify the overall performance boost achieved, where gray denotes the baseline dataset and green denotes the overall highest performance gain on all metrics over several different models.

For the optimizations in Phase 2, we use a simple multi-layer perceptron for initial evaluation. The final configuration of each stage is then tested on Random Forest, AdaBoost, XGBoost, MLP, PointNet++ and PointNext-b to get a robust estimation for the impact of the specific configuration. The ALS dataset consists of 12 manually labeled point clouds, where segments 0–9 are used for training and segments 10 and 11 are used for validation. Though this dataset is very small, we show that through our preprocessing regiment, the models are still able to generalize to new data. We labeled 3 additional point cloud segments that were not involved



in any of the previous steps as test data. After determining the best strategy on the ALS data, we apply the same workflow to the MMS dataset and evaluate the performance boost on three of those segments. We structured the evaluation in a way that we can compare different configurations directly to being able to interpret the results of each consecutive step. Improvements are reported with respect to the models trained on stage 2.1, since comparing different data foundations would be pointless. Configuration 2.1 only employs Step 1 and does not consider ground filtering. In configuration 3.1, we evaluate the impact of treating ground and non-ground points separately. In configuration 4.1, we use the full scheme of Phase 1. We split each stage into two leaves, employing our Phase 2, consisting of feature extraction and selection, in the second leaf while leaving it out in leaf one. This procedure allows for a detailed evaluation of the different steps, leading to a better understanding of the differences in data as well as learning methods. We train a PointNeXt-b model for validation of the performanceboosting potential on more recent architectures in Stages 2.1, 2.2, 3.2, and 4.2.

5 Results

In this section, we present the results of our comparison of the different stages of our preprocessing workflow. The results of our ablation study showed that using optimal neighborhood outperforms manual receptive field optimization while significantly

reducing the computation time. Additionally, the measures taken to make the feature extraction self-adaptive with respect to the data distribution based on the optimal neighborhood approach benefit the degree of automation of the workflow. We compare the results of our workflow evaluation strategy based on overall accuracy (OA) (Equation 4), precision (pr) (Equation 5) and recall (rec) (Equation 6), weighted mean intersection over union (mIoU) (w) (Equations 7, 8) and unweighted intersection over union (unw. mIoU) (u) (Equation 9) to better measure performance changes for minority classes. Those metrics are calculated by first counting true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) and applying the following formulars, where n is the number of samples, i denotes the class and N denotes the number of classes. The difference between weighted mIoU and unweighted mIoU is that the number of samples per class affects the weighted metric; therefore, in cases where there is a large class imbalance, the unweighted IoU better captures improvements of minority classes, while the weighted mIoU better measures the overall performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
(4)

$$Precision = \frac{TP}{TP + FP}$$
(5)

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$IoU = \frac{TP}{TP + FN + FP}$$
(7)

$$mIoU = \frac{\sum_{i} n_i * IoU_i}{\sum_{i} n_i} \tag{8}$$

$$unweighted \ mIoU = \frac{\sum_{i} IoU_{i}}{N}$$
(9)

5.1 Results - preprocessing workflow

In this section, we discuss the results of our six workflow configurations on the six chosen models. The results for the full feature set are shown in Table 2. For PointNet++, we reduced the input point number to 2048 points due to the low point density of the ALS dataset in configurations 2.1 and 2.2, to 1,024 points for ground points in 3.1, 3.2, 4.1, and 4.2, and to just 512 points for above ground points. The block size was extended to 10 m for all configurations. 4,096 points were used for the MMS data. The remaining major hyperparameters of PointNet++ were chosen according to the default values:

- Learning rate 0.001
- Step size: 10
- Optimizer: Adam
- Learning rate decay: 0.7
- Weight decay rate: 0.0001

The model was trained for a maximum of 200 epochs and varying batch sizes due to the large inputs for full feature sets. The batch size ranged from 64 for the 2.1, 3.1, and 4.1 configuration down to 8 for the 2.2, 3.2, and 4.2 configuration.

For our future target application of model generation, the most important classes are: road, non-road ground, road markings and road furniture. Since we focus on the road structure, the classes "vehicle" and "vegetation" mainly fulfill the purpose of separating the important classes from the less important objects in the road environment. For our PointNeXt model, we chose the default configuration, except for the voxel sampling size of 0.04 and an initial radius of 0.3 compared to the default of 0.1. We used the relatively light-weight PointNeXt-b model for fast training time. The number of points per class in the training dataset are significantly imbalanced where the majority class (road) has approximately 3 million points whereas the minority class (road furniture) only comprises 63,000 points. The number of points per class for the ALS dataset are denoted beneath the class names in Table 2, whereas those for the MMS dataset are shown in 3. The majority class of the MMS dataset is the road class, while the minority class is the non road ground class. The class imbalance is representative for validation and test sets for both datasets. No additional measures for compensating class imbalance were taken during the training of any of the models.

To clarify the significant improvements over the stages, we elaborate the performance differences between stages at the example of PointNeXt. Starting with the baseline dataset (highlighted in gray) the most significant performance metrics are bold. Adding the Phase 2 preprocessing (2.2) (orange), the overall accuracy increases by +9.70%, while unweighted mIoU increases sharply by +21.49%. Adding ground separation in stage 3.2 (yellow), while only adding +0.97% in overall accuracy, stage 2.2 is strongly outperformed regarding mIoU with a performance gain of additional +11.04%

unweighted mIoU. This performance is matched and partially outperformed in stage 4.2 (green) when using the road block approach, adding another +2.68% to the overall accuracy and increasing the weighted mIoU by +5.08%, while only slightly losing on unweighted mIoU (-0.13%) compared to stage 3.2. This consistently increasing performance throughout the steps of our approach is mirrored by all other models as well and totals a performance gain against the baseline of +13,35% OA, +15.16% weighted mIoU and +32.40% in unweighted mIoU for PointNeXt. Additionally, if only the stages without feature extraction are taken into comparison (2.1, 3.1, and 4.1), the models still show a consistent performance gain, in unweighted mIoU, e.g., for Random Forest with 16.90% from 2.1 to 3.1 and another 6.80% from 3.1 to 4.1 (highlighted bold). In a way, one could argue that the steps involving only the first phase of preprocessing are data cleaning and homogenization steps, which enable better feature aggregation. Limiting receptive fields in the edge areas of road blocks and adding small error portions through erroneously separated points during ground filtering corresponds to lowering the upper boundary of the achievable performance. However, these steps add a small bias, which averaged at approximately 2% of points in the ALS dataset, compared to the significant boost of model performances when applying feature extraction to their resulting point cloud subset.

It can be noted that the results for the machine learning models on stage 2.2 could be significantly improved. Specifically, the minimum performance gain over all four machine learning models (MLP, Random Forest, XGBoost, AdaBoost) exceeded 11.4% of unweighted mIoU for the XGBoost model, while Random Forest improved by 27,25% in the same metric comparing 2.1 to 2.2, even surpassing PointNet++ performance of stage 2.2 by 4.69% unweighted mIoU. PointNet++ shows no significant change in performance until stage 3.2, where the unweighted mIoU improved by 10.87% compared to stage 2.1. It becomes obvious that on the ALS dataset, the PointNeXt model easily outperforms PointNet++ in stages with feature generation.

Generally, we assume that the low sensitivity of PointNet++ to the preprocessing workflow on the ALS dataset is due to the low point density, leading to low variability in data when using the blockwise data input approach of PointNet++. Here it becomes obvious that the change made in PointNett, directly running on the full scene similar to Randla-Net (Hu et al., 2020), is better suited for sparse data than sampling blocks with a fixed number of points in them. We validate this assumption by comparing results of the ALS dataset with those of the MMS dataset, which provides a higher point density in the next section.

5.2 Results - MMS dataset comparison

The MMS dataset was first randomly subsampled the sections to a point density of 120 points per square meter. The object classes differed slightly compared to the ALS dataset lacking the car class, since dynamic objects had been removed beforehand. The individual sections covered approximately the same lateral area as the ALS data after step 1 of our workflow and were not georeferenced. We, therefore, focus on evaluating the impact of our phase 2 for the comparison, demonstrating the flexibility of our pipeline. Since

St.	Mod.	OA	w	u	N. R.	Gr.	Road		R. Mark.		Veg.		Car		R. Frunit.	
			Num	points	2166	5946	3057	3057436		51	7663	310	109877		63069	
					Pr	Rec	Pr	Rec	Pr	Rec	Pr	Rec	Pr	Rec	Pr	Rec
	MLP	71,45	55,58	31,2	0.73	0.81	0.74	0.94	0.38	0.12	0.81	0.48	0.08	0.01	0	0
	RF	79,91	66,54	36,0	0.78	0.8	0.86	0.93	0.64	0.28	0.75	0.83	0.34	0	0	0
2.1	AB	74,48	59,35	31,4	0.74	0.82	0.72	0.92	0.17	0.0	0.93	0.55	0.26	0.04	1.00	0
(baseline)	XGB	70,35	54,26	35,6	0.77	0.72	0.75	0.84	0.67	0.43	0.72	0.53	0.08	0.47	0.46	0.14
	PN++	88,65	76,49	58,55	0.9	0.7	0.88	1.00	0.79	0.56	0.88	0.7	0.76	0.57	0.68	0.34
	PNX	82,53	76,61	48,40	0.85	0.97	0.87	0.97	0.57	0.00	0.94	1.00	0.71	0.04	0.99	0.24
	MLP	83,86	72,22	52,19	0.89	0.78	0.88	0.94	0.58	0.61	0.6	0.97	0.81	0.49	0.30	0.19
	RF	87,83	78.30	63,25	0.93	0.85	0.87	0.98	0.68	0.51	0.96	0.91	0.79	0.65	0.33	0.76
2.2	AB	84,87	71,68	49,04	0.91	0.87	0.93	0.96	0.52	0.68	0.75	0.46	0.65	0.53	0.1	0.68
2.2	XGB	85,67	74,93	47,01	0.9	0.82	0.92	0.95	0.54	0.71	0.83	0.88	0.72	0.36	0.19	0.61
	PN++	88,39	76,25	58,56	0.81	0.95	0.97	0.91	0.71	0.5	0.98	0.97	0.57	0.53	0.12	0.50
	PNX	92,23	84,72	69,89	0.86	0.86	0.95	0.98	0.94	0.76	0.99	0.68	0.66	0.50	0.85	0.94
	MLP	78,64	64,75	45,26	0.87	0.72	0.75	0.95	0.47	0.12	0.9	0.78	0.68	0.87	0.48	0.01
	RF	81,26	68,48	53,1	0.83	0.77	0.92	0.92	0.35	0.51	0.84	0.87	0.76	0.74	0	0
3.1	AB	79,94	67,63	52,82	0.81	0.88	0.92	0.91	0.66	0.31	0.76	0.64	0.67	0.67	0	0
	XGB	78,49	65,41	51,56	0.83	0.81	0.88	0.93	0.47	0.36	0.84	0.55	0.68	0.79	0.05	0.12
	PN++	88,99	70,35	54,97	0.86	0.88	0.95	0.94	0.69	0.69	0.96	0.98	0.44	0.17	0.49	0.03
	MLP	84,85	73,73	65,08	0.94	0.59	0.90	0.98	0.48	0.69	0.95	0.89	0.57	0.55	0.63	0.92
	RF	92,91	86,81	81,10	0.89	0.87	0.96	0.98	0.78	0.71	0.99	0.94	0.77	0.89	0.79	0.96
2.2	AB	89,85	81,57	70,79	0.9	0.89	0.91	0.96	0.78	0.55	0.98	0.91	0.84	0.95	0.74	0.47
5.2	XGB	91,38	84,13	71,75	0.89	0.94	0.95	0.96	0.78	0.49	0.98	0.91	0.86	0.91	0.55	0.68
	PN++	86,96	81,48	69,42	0.89	0.84	0.96	0.93	0.60	0.81	0.81	0.99	0.9	0.38	0.88	0.93
	PNX	93,20	86,69	80,93	0.92	0.83	0.97	0.99	0.81	0.88	1.00	0.70	0.81	0.97	0.92	0.96
	MLP	87,62	78,09	61,55	0.9	0.9	0.89	0.98	0.89	0.3	0.89	0.92	0.82	0.77	0.32	0.23
	RF	87,2	77,31	59,9	0.92	0.86	0.86	0.98	0.7	0.3	0.89	0.93	0.86	0.79	0.56	0.59
4.1	AB	85,64	74,88	61,83	0.92	0.79	0.85	0.98	0.56	0.46	0.95	0.89	0.83	0.8	0.41	0.76
	XGB	86,96	76,97	62,57	0.92	0.86	0.88	0.98	0.59	0.4	0.94	0.91	0.86	0.72	0.33	0.87
	PN++	87,31	76,06	61,32	0.86	0.9	0.98	0.91	0.58	0.68	0.77	0.68	0.9	0.84	0.23	0.59

TABLE 2 Results of our evaluation strategy for all used models on ALS data.

(Continued on the following page)

St.	Mod.	OA	w	u	N. R. Gr.		Road		R. Mark.		Veg.		Car		R. Frunit.		
			Num	points	2166	2166946		3057436		387451		766310		109877		63069	
					Pr	Rec	Pr	Rec	Pr	Rec	Pr	Rec	Pr	Rec	Pr	Rec	
	MLP	86,55	76,35	65,15	0.88	0.69	0.92	0.98	0.54	0.63	0.98	0.89	0.50	0.70	0.58	0.62	
	RF	92,35	85,82	80,38	0.86	0.89	0.95	0.98	0.86	0.65	0.99	0.93	0.71	0.9	0.84	0.96	
	AB	88,37	79,17	70,57	0.88	0.86	0.91	0.96	0.63	0.48	0.96	0.89	0.82	0.94	0.89	0.62	
4.2	XGB	89,57	81,11	69,97	0.93	0.86	0.95	0.96	0.53	0.67	0.95	0.9	0.85	0.86	0.52	0.77	
	PN++	85,07	71,05	61,82	0.82	0.95	0.98	0.91	0.74	0.61	0.85	0.81	0.48	0.59	0.78	0.54	
	PNX	95,88	91,77	80,80	0.93	0.87	0.96	1.00	0.94	0.82	0.99	0.99	0.92	0.77	0.62	0.91	

TABLE 2 (Continued) Results of our evaluation strategy for all used models on ALS data.

Random Forest, MLP, PointNet++ and PointNeXt had previously shown to be most interesting for further comparison, we tested only with these four models.

From the results in Table 3, it becomes obvious, that the higher point density has a positive effect on the performance of PointNet++ and PointNeXt. It can be noted, that in comparison to the ALS dataset, where the intensity is much noisier than in the MMS dataset, the road markings were far better extracted, while the distinction between road surface and the lateral non-road area remained the main challenge for models trained on the MMS dataset. The performance of PointNet++ strongly increases over each consecutive stage, more than doubling the unweighted mIoU between stage 1.1 and stage 3.2, adding over 43%. PointNeXt delivered strong performance even in Stage 1.1, without any preprocessing. Still, the feature extraction in Stage 2.2 could improve the unweighted mIoU by over 5%, while additionally adding 5.62% in overall accuracy. Another interesting finding being demonstrated through the MMS dataset is that the machine learning models could be strongly improved, more than doubling unweighted mIoU and adding more than 20% overall accuracy in the case of Random Forest from stage 1 to stage 2.2 and over 14% for MLP. Still, both were significantly outperformed by the deep learning models in stage 3.2 regarding unweighted mIoU.

5.3 Results - feature selection

After demonstrating the performance enhancement potential of our work, we want to briefly outline the potential for reducing the computation time demand of our workflow, focusing on training time in this subsection. It is to show how the model performance changes when using feature subsets assembled via our GA. For the sake of computation speed, we chose a simple naive Bayesian classifier for validation with the unweighted mIoU as decision metric. We compared four different numbers of features 20, 35, 55 and the full feature set for each of the five datasets from configurations 2.2, 3.2 ground, 3.2 non-ground, 4.2 ground and 4.2 non-ground. The results of the comparison are depicted in Figure 9. We, additionally, used the resulting feature subsets to train an MLP for a second comparison, which is depicted in Figure 10. We used a fixed model configuration for the MLP, specifying the following parameters:

- Hidden layer size: 500
- Activation: ReLu
- Optimizer: Adam
- Maximum iteration: 5

Comparing the two different classifiers, the Bayesian classifier seems to improve for smaller feature sets, which might be the case due to it assuming uncorrelated features while many of the extracted features are in fact correlated. Overall, it can be noted that our feature selection scheme works well in reducing feature space while maintaining performance. The comparison between the datasets of the different stages also indicates the performance boost of each consecutive stage of our workflow. The performance of the Bayesian classifier additionally shows that through our automatic workflow, even simple models become applicable to the segmentation task.

With these results, it becomes clear that, while we are extracting a high number of features in every stage, the amount of information we encode into individual features may still increase in higher stages. This might be due to improved receptive fields with reduced heterogeneity and more characteristic neighborhoods being aggregated. This interesting finding could lead to a significant reduction in precomputation effort by introducing a learning scheme for feature relevance directly during computation. The potential future research topics will be discussed in more detail in Section 6.

6 Discussion

Through our extensive evaluation of the proposed preprocessing workflow, we could show that it significantly boosts performance on versatile data foundations. Sparse as well as dense point cloud data in the road environment is consistently enriched and thereby far better suited for arbitrary machine learning and deep learning models. A notable achievement is the finding that enhancing point

St.	Mod.	OA	w	u	N. R. Gr. 2343695		Road		R. Ma	rk.	Veg.		R. Frunit.		
			Num	points			4066	6973	19531	186	34159	971	5968978		
					Pr	Rec	Pr	Rec	Pr	Rec	Pr	Rec	Pr	Rec	
1	MLP	78,16	64,16	33,51	0.00	0.00	0.87	0.97	0.39	0.27	0.42	0.27	0.62	0.63	
	RF	74,94	59,92	28,67	0.00	0.00	0.84	0.93	0.23	0.07	0.65	0.23	0.49	0.64	
	PN++	61,64	44,55	42,54	0.44	0.06	0.78	0.67	0.13	1.00	0.93	0.67	0.78	0.95	
	PNX	90.66	93.33	85.12	0.87	0.84	1.00	0.99	0.97	1.00	0.74	0.85	0.94	0.96	
	MLP	92,20	85,52	66,96	0.19	0.63	0.99	0.94	0.76	0.84	0.86	0.84	0.90	0.93	
	RF	95,65	91,67	66,72	0.98	0.05	0.97	1.00	0.99	0.68	0.84	0.90	0.94	0.92	
2.2	PN++	94,74	90,01	66,43	0.62	0.93	0.97	0.99	0.95	0.60	1.00	0.25	0.94	0.99	
	PNX	96.04	97.52	90.29	0.94	0.90	0.99	1.00	0.93	0.83	0.97	0.96	0.97	0.98	
	MLP	86,66	76,72	59,81	0.72	0.56	0.92	0.97	0.70	0.43	0.71	0.72	0.86	0.85	
3.2	RF	88,24	79,06	60,84	0.92	0.44	0.90	1.00	0.87	0.37	0.73	0.84	0.91	0.84	
	PN++	93,26	88,05	86,01	0.95	0.60	0.91	0.99	0.92	1.00	1.00	0.99	0.95	0.98	
	PNX	97.23	94.42	87.59	0.96	0.97	0.97	0.99	0.97	0.57	0.98	0.96	0.96	0.98	

TABLE 3 Results of stages 1, 2.2 and 3.2 on MMS dataset comparing PointNet++, PointNeXt-b, MLP and Random Forest.



cloud data in this way makes models more receptive to imbalanced class distributions, which otherwise limits the applicability of unspecialized machine learning approaches. While the sparsity may still be a factor for specific models, working on fixed block sizes to sample input data, models that mitigate this issue by using either full scenes to generate input or even work with point-wise input take a large profit of our workflow. Therefore, the main limitation of our approach is the computation time needed for dataset preprocessing. We measured the computation time for the processing of Stage 2 of the ALS dataset averaging to 53 min per 1 million points, which results in a total processing time of approximately 6 h and 30 min (GPU: Nvidia Quadro 5000, CPU: Intel i9-11900K, RAM: 64 GB). It has to be mentioned that the preprocessing at the current stage is purely Python-based and will be accelerated in the future by moving to the C++ programming language. This aligns with the strategies of many point cloud processing



neural network architectures, where preprocessing frameworks are typically implemented in C++ to accelerate the steps before model training as done in (Qian et al., 2022) with their openpoints library. However, our aim is the optimization of semantic segmentation for further use in geometric-semantic model generation. For geometricsemantic model generation from segmented point clouds, it is essential to lay a robust and as accurate as possible semantic foundation in order to enable automatic and sound object generation in the next steps. We therefore accept the longer computation time if the quality of the predicted results increases significantly, which we have proven. The possibility of performing feature extraction on arbitrary data with differing semantic classes simplifies the process of machine learning, since we can improve the models' robustness to class imbalances while also greatly improving the overall model performance. The modular approach allows for easy customization of the workflow to georeferenced data and data in local coordinate reference systems. In that way, incorporating our preprocessing workflow into deep learning models to improve segmentation results even leads to a reduction in the amount of data necessary. We have gained insights into the change in model performance when changing the receptive fields in feature extraction, and could thereby derive new research ideas towards direct integration of prior knowledge into deep neural network architectures.

7 Conclusion

With our work, we could show that our workflow improves the performance of arbitrary machine learning and deep learning models on ALS as well as on MMS datasets of the road space. Our modular and flexible preprocessing workflow is applicable to arbitrary neural network architectures and is capable of mitigating class imbalances and limited data supply. The best performing model regarding mIoU for the MMS dataset was the most recent PointNeXt model, which shows that even state-of-the-art neural networks can be boosted by data enrichment to date.

Following our findings from the immaculate evaluation of our workflow, we can answer RQ1, since we established a workflow directly applicable to arbitrary data, as we showed through evaluation of both ALS and MMS datasets. With our geospatial data integration step, we are able to automatically and efficiently limit the area of interest to the road space and further into homogenous road blocks, while data in local coordinate reference systems can still be processed with steps 3 and step 4. This, together with the finding that refining data subsets via ground filtering and road block extraction leads to an improved capability of extracting meaningful features with our approach, answers our RQ2. The combination of our four steps showed consistent improvement on all models, with the exception of PointNet++ in the ALS case, where the data sparsity negatively interacted with the data processing scheme of the deep learning architecture. Still, given this caveat, being a solvable problem as demonstrated by comparison between PointNet++ and its successor PointNeXt, where one significant change is the approach for processing input data, we consider RQ3 solved. Lastly, the optimal neighborhood approach we intertwined into the majority of our feature set, effectively allowed the process to automatically adapt to changing data. We additionally analyzed, that this approach not only automates the adaptation of receptive fields for feature extraction, but even matches the performance of tedious manual optimization of the receptive fields for several sets of features in the ablation study. We, therefore, also consider RQ4 answered.

Since in this work we focused on highway environments only, it is yet to be evaluated whether our approach can also improve the performance of point cloud semantic segmentation for different target environments, such as indoor semantic segmentation or bridge semantic segmentation. In future work, we aim to directly incorporate our preprocessing workflow into artificial neural network architectures to improve existing model architectures and to allow out-of-the-box performance boosting, without the necessity of excessive precomputation. This will be achieved by directly learning the most information-rich features currently handled in our feature selection step, which showed that the performance boost

can be maintained well by choosing the most representative features from the full feature set. This will accelerate the processing time on the one hand and make the preprocessing scheme even more adaptive to changing datasets on the other hand. Alternatively, our road block approach could potentially be deployed for blockbased neural networks based on PointNet to directly utilize our road blocks to gain context information by an ordered reception over roads. Another idea is to combine the roadblock approach with image data of the road surface to incorporate more knowledge of the pavement. Generally, image data is better suited for crack detection, and mapping those detected damages into the point cloud would allow the generation of more realistic geometricsemantic as-is models for damage assessment within a digital twin use case. The workflow could also be extended to buildings or land use applications in the future, since geospatial data of the same kind we used is available for different domains and could be applied similarly. This will allow the evaluation of the performance enhancement potential for more diverse scenes, apart from the road space we focused on in this contribution. Overall, improving the semantic segmentation quality within the process chain of Scan-to-Twin is essential to creating correct and accurate digital representations for digital twins. The value of a digital twin depends on the accuracy of its predictions and implications. Therefore, the errors and uncertainties within the base model have to be minimized and, if possible, quantified to allow the system to become a reliable decision-support. In our future work, we will focus on completing the comprehensive framework to automatically derive and control the geometric-semantic models for digital twins of roads.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation. The datasets will be made available under: https://github.com/DavidCrampen/Dataenrichment-for-semantic-segmentation-of-point-clouds-of-roads.

Author contributions

DC: Visualization, Writing – original draft, Formal Analysis, Conceptualization, Investigation, Methodology, Data curation, Software, Validation. JB: Supervision, Project administration, Writing – review and editing, Funding acquisition, Resources.

References

Alexiou, E., Zhou, X., Viola, I., and Cesar, P. (2024). Pointpca: point cloud objective quality assessment using pca-based descriptors. *EURASIP J. Image Video Process.* 20. doi:10.48550/arXiv.2111.12663

Aljumaily, H., Laefer, D. F., and Cuadra, D. (2019). Integration of lidar data and gis data for point cloud semantic enrichment at the point level. *Photogrammetric Eng. and Remote Sens.* 85, 29–42. doi:10.14358/PERS.85.1.29

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., et al. (2016). "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1534–1543. doi:10.1109/CVPR. 2016.170

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work has been conducted for and funded by the Collaborative Research Center SFB/TRR 339 "Digital Twin of the Road" (Project number 453596084). We acknowledge the funding by the German Research Foundation (DFG).

Acknowledgments

We gratefully acknowledge Straßen.NRW (Landesbetrieb Straßenbau Nordrhein-Westfalen) for providing access to their data.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbuil.2025. 1607375/full#supplementary-material

Atik, M. E., and Duran, Z. (2022). Selection of relevant geometric features using filter-based algorithms for point cloud semantic segmentation. *Electronics* 11, 3310. doi:10.3390/electronics11203310

Baligh Jahromi, A., Zoej, M. J. V., Mohammadzadeh, A., and Sadeghian, S. (2011). A novel filtering algorithm for bare-earth extraction from airborne laser scanning data using an artificial neural network. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* 4, 836–843. doi:10.1109/JSTARS.2011.2132793

Besl, P. J., and Jain, R. C. (1988). Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Analysis Mach. Intell.* 10, 167–192. doi:10.1109/34.3881

Bhanu, B., Lee, S., Ho, C.-C., and Henderson, T. (1986). "Range data processing: representation of surfaces by edges," in Proceedings of the eighth international conference on pattern recognition (Piscataway, NJ, USA: IEEE Computer Society Press), 238.

Biçici, S., and Zeybek, M. (2021). Effectiveness of training sample and features for random forest on road extraction from unmanned aerial vehicle-based point cloud. *Transp. Res. Rec.* 2675, 401–418. doi:10.1177/03611981211029645

Boulch, A., Saux, B. L., and Audebert, N. (2017). "Unstructured point cloud semantic labeling using deep segmentation networks," in *Proceedings of the workshop on 3D object retrieval* (Lyon, France: Goslar, DEU: Eurographics Association), 17–24. doi:10.2312/3dor.20171047

Bremer, M., Wichmann, V., and Rutzinger, M. (2013). Eigenvalue and graph-based object extraction from mobile laser scanning point clouds. *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.* II-5 (W2), 55–60. doi:10.5194/isprsannals-II-5-W2-55-2013

Chen, C., Guo, J., Wu, H., Li, Y., and Shi, B. (2021). Performance comparison of filtering algorithms for high-density airborne lidar point clouds over complex landscapes. *Remote Sens.* 13, 2663. doi:10.3390/rs13142663

Crampen, D., and Blankenbach, J. (2024). Semantic segmentation of road infrastructure point clouds supported by geospatial data. *arXiv*, 4951547. doi:10.2139/ssrn.4951547

Crampen, D., Hein, M., and Blankenbach, J. (2024). "A level of as-is detail concept for digital twins of roads—a case study," in *Recent advances in 3D geoinformation science*. Editors T. H. Kolbe, A. Donaubauer, and C. Beil (Cham: Springer Nature Switzerland), 499–515.

Esfahani, M. E., Rausch, C., Sharif, M. M., Chen, Q., Haas, C., and Adey, B. T. (2021). Quantitative investigation on the accuracy and precision of scanto-bim under different modelling scenarios. *Automation Constr.* 126, 103686. doi:10.1016/j.autcon.2021.103686

Federal Ministry for Digital and Transport, BMDV Germany (2021). Masterplan bim bundesfernstraßen. Available online at: https://bmdv.bund.de/ SharedDocs/DE/Anlage/StB/bim-rd-masterplan-bundesfernstrassen.pdf?__blob=pub licationFile (Accessed March 26, 2025).

Grandio, J., Riveiro, B., Soilán, M., and Arias, P. (2022). Point cloud semantic segmentation of complex railway environments using deep learning. *Automation Constr.* 141, 104425. doi:10.1016/j.autcon.2022.104425

Grilli, E., Menna, F., and Remondino, F. (2017). A review of point clouds segmentation and classification algorithms. *Int. Archives Photogrammetry, Remote Sens. Spatial Inf. Sci.* XLII-2 (W3), 339–344. doi:10.5194/isprs-archives-XLII-2-W3-339-2017

Hirt, P.-R., Hoegner, L., and Stilla, U. (2021). A concept for the segmentation of individual urban trees from dense mls point clouds. *Int. Archives Photogrammetry, Remote Sens. Spatial Inf. Sci.* XLIII-B2-2021, 171–178. doi:10.5194/isprs-archives-XLIII-B2-2021-171-2021

Hu, Q., Liu, D., and Hu, W. (2023). Density-insensitive unsupervised domain adaption on 3d object detection. *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 17556–17566. doi:10.1109/cvpr52729.2023.01684

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., et al. (2020). Randla-net: efficient semantic segmentation of large-scale point clouds. *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 11108–11117. doi:10.1109/CVPR42600.2020.01112

Huang, Y., and Li, L. (2011). Naive bayes classification algorithm based on small sample set. 2011 IEEE Int. Conf. Cloud Comput. Intell. Syst., 34–39. doi:10.1109/CCIS.2011.6045027

Interministerial Committee (2025). Interministerial committee for spatial data infrastructures in North Rhine-Westphalia. *OpenGeoPortalNRW*.

Justo, A., Soilán, M., Sánchez-Rodríguez, A., and Riveiro, B. (2021). Scan-to-bim for the infrastructure domain: generation of ifc-compliant models of road infrastructure assets and semantics using 3d point cloud data. *Automation Constr.* 127, 103703. doi:10.1016/j.autcon.2021.103703

Li, F., Lehtomäki, M., Oude Elberink, S., Vosselman, G., Kukko, A., Puttonen, E., et al. (2019a). Semantic segmentation of road furniture in mobile laser scanning data. *ISPRS J. Photogrammetry Remote Sens.* 154, 98–113. doi:10.1016/j.isprsjprs.2019.06.001

Li, F., and Vosselman, G. (2018). Pole-like road furniture detection and decomposition in mobile laser scanning data based on spatial relations. *Remote Sens.* 10, 531. doi:10.3390/rs10040531

Li, Y., Wang, W., Li, X., Xie, L., Wang, Y., Guo, R., et al. (2019b). Pole-like street furniture segmentation and classification in mobile lidar data by integrating multiple shape-descriptor constraints. *Remote Sens.* 11, 2920. doi:10.3390/rs11242920

Lucas, C., Bouten, W., Koma, Z., Kissling, W., and Seijmonsbergen, A. (2019). Identification of linear vegetation elements in a rural landscape using lidar point clouds. *Remote Sens.* 11, 292. doi:10.3390/rs11030292

Ma, H., Ma, H., Zhang, L., Liu, K., and Luo, W. (2022). Extracting urban road footprints from airborne lidar point clouds with pointnet++ and two-step post-processing. *Remote Sens.* 14, 789. doi:10.3390/rs14030789

Ma, H., Pei, Z., Wei, Z., and Zhong, R. (2017). Automatic extraction of road markings from mobile laser scanning data. *Int. Archives Photogrammetry, Remote Sens. Spatial Inf. Sci.* XLII-2 (W7), 825–830. doi:10.5194/isprs-archives-XLII-2-W7-825-2017

Mansour, M., Martens, J., and Blankenbach, J. (2024). Hierarchical svm for semantic segmentation of 3d point clouds for infrastructure scenes. *Infrastructures* 9, 83. doi:10.3390/infrastructures9050083

Martens, J., and Blankenbach, J. (2023). Vox2bim+ - a fast and robust approach for automated indoor point cloud segmentation and building model generation. *PFG – J. Photogrammetry, Remote Sens. Geoinformation Sci.* 91, 273–294. doi:10.1007/s41064-023-00243-1

Murtiyoso, A., and Grussenmeyer, P. (2019). Point cloud segmentation and semantic annotation aided by gis data for heritage complexes. *Int. Archives Photogrammetry, Remote Sens. Spatial Inf. Sci.* XLII-2/W9, 523–528. doi:10.5194/isprs-archives-XLII-2-W9-523-2019

Murtiyoso, A., Veriandi, M., Suwardhi, D., Soeksmantono, B., and Harto, A. (2020). Automatic workflow for roof extraction and generation of 3d citygml models from low-cost uav image-derived point clouds. *ISPRS Int. J. Geo-Information* 9, 743. doi:10.3390/ijgi9120743

Noroozinejad Farsangi, E., Shehata, A. O., Rashidi, M., Ghassempour, N., and Mirjalili, S. (2024). Transitioning from bim to digital twin to metaverse. *Front. Built Environ.* 10, 1486423. doi:10.3389/fbuil.2024.1486423

Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2002). Shape distributions. ACM Trans. Graph. 21, 807–832. doi:10.1145/571647.571648

Paden, I., Garcia-Sanchez, C., and Ledoux, H. (2022). Towards automatic reconstruction of 3d city models tailored for urban flow simulations. *Front. Built Environ.* 8, 899332. doi:10.3389/fbuil.2022.899332

Park, Y., and Guldmann, J.-M. (2019). Creating 3d city models with building footprints and lidar point cloud classification: a machine learning approach. *Comput. Environ. Urban Syst.* 75, 76–89. doi:10.1016/j.compenvurbsys.2019.01.004

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: deep learning on point sets for 3d classification and segmentation. *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 652–660. doi:10.1109/CVPR.2017.16

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: deep hierarchical feature learning on point sets in a metric space. *Adv. neural Inf. Process. Syst.* 30, 14. doi:10.48550/arXiv.1706.02413

Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., et al. (2022). Pointnext: revisiting pointnet++ with improved training and scaling strategies. *Adv. neural Inf. Process. Syst.* 35, 23192–23204. doi:10.48550/arXiv.2206.04670

Rato, D., and Santos, V. (2021). Lidar based detection of road boundaries using the density of accumulated point clouds and their gradients. *Robotics Aut. Syst.* 138, 103714. doi:10.1016/j.robot.2020.103714

Roynard, X., Deschaud, J.-E., and Goulette, F. (2018a). Classification of point cloud for road scene understanding with multiscale voxel deep network. *10th workshop Plan. Perceptionand Navigation Intelligent Veh. PPNIV'2018*, 1–11. doi:10.48550/arXiv.1804.03583

Roynard, X., Deschaud, J.-E., and Goulette, F. (2018b). Paris-lille-3d: a large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int. J. Robotics Res.* 37, 545–557. doi:10.1177/0278364918767506

Rusu, R. B., Blodow, N., and Beetz, M. (2009). Fast point feature histograms (fpfh) for 3d registration. 2009 IEEE Int. Conf. Robotics Automation (IEEE), 3212–3217. doi:10.1109/ROBOT.2009.5152473

Safaie, A. H., Rastiveis, H., Shams, A., Sarasua, W. A., and Li, J. (2021). Automated street tree inventory using mobile lidar point clouds based on hough transform and active contours. *ISPRS J. Photogrammetry Remote Sens.* 174, 19–34. doi:10.1016/j.isprsjprs.2021.01.026

Schatz, Y., and Domer, B. (2024). Semi-automated creation of ifc bridge models from point clouds for maintenance applications. *Front. Built Environ.* 10, 1375873. doi:10.3389/fbuil.2024.1375873

Schlosser, A. D., Szabó, G., Bertalan, L., Varga, Z., Enyedi, P., and Szabó, S. (2020). Building extraction using orthophotos and dense point cloud derived from visual band aerial imagery based on machine learning and segmentation. *Remote Sens.* 12, 2397. doi:10.3390/rs12152397

Shi, X., Ma, H., Chen, Y., Zhang, L., and Zhou, W. (2018). A parameter-free progressive tin densification filtering algorithm for lidar point clouds. *Int. J. Remote Sens.* 39, 6969–6982. doi:10.1080/01431161.2018.1468109

Shin, Y.-H., Son, K.-W., and Lee, D.-C. (2022). Semantic segmentation and building extraction from airborne lidar data with multiple return using pointnet++. *Appl. Sci.* 12, 1975. doi:10.3390/app12041975

Sithole, G., and Vosselman, G. (2001). Filtering of laser altimetry data using a slope adaptive filter. *Int. Archives Photogrammetry Remote Sens. Spatial Inf. Sci.* 34, 203–210.

Soilán, M., Riveiro, B., Martínez-Sánchez, J., and Arias, P. (2017). Segmentation and classification of road markings using mls data. *ISPRS J. Photogrammetry Remote Sens.* 123, 94–103. doi:10.1016/j.isprsjprs.2016.11.011

Tan, F., Fu, X., Zhang, Y., and Bourgeois, A. G. (2007). A genetic algorithm-based method for feature subset selection. *Soft Comput.* 12, 111–120. doi:10.1007/s00500-007-0193-8

Vassilev, H., Laska, M., and Blankenbach, J. (2024). Uncertainty-aware point cloud segmentation for infrastructure projects using bayesian deep learning. *Automation Constr.* 164, 105419. doi:10.1016/j.autcon.2024.105419

Wang, B., Lan, J., and Gao, J. (2022). Lidar filtering in 3d object detection based on improved ransac. *Remote Sens.* 14, 2110. doi:10.3390/rs14092110

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph. (tog)* 38, 1–12. doi:10.1145/3326362

Wang, Z., Yang, L., Sheng, Y., and Shen, M. (2021). Pole-like objects segmentation and multiscale classification-based fusion from mobile point clouds in road scenes. *Remote Sens.* 13, 4382. doi:10.3390/rs13214382

Weinmann, M., Jutzi, B., and Mallet, C. (2014). Semantic 3d scene interpretation: a framework combining optimal neighborhood size selection with relevant features. *ISPRS Ann. Photogrammetry, Remote Sens. Spatial Inf. Sci.* II-3, 181–188. doi:10.5194/isprsannals-II-3-181-2014

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., et al. (2015). 3d shapenets: a deep representation for volumetric shapes. *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 1912–1920. doi:10.48550/arXiv.1406.5670

Xu, J., Zhang, R., Dou, J., Zhu, Y., Sun, J., and Pu, S. (2021). Rpvnet: a deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 16024–16033. doi:10.48550/arXiv.2103.12978

Yan, Y., and Hajjar, J. F. (2021). Automated extraction of structural elements in steel girder bridges from laser point clouds. *Automation Constr.* 125, 103582. doi:10.1016/j.autcon.2021.103582

Yang, R., Li, Q., Tan, J., Li, S., and Chen, X. (2020). Accurate road marking detection from noisy point clouds acquired by low-cost mobile lidar systems. *ISPRS Int. J. Geo-Information* 9, 608. doi:10.3390/ijgi9100608

Yilmaz, V. (2021). Automated ground filtering of lidar and uas point clouds with metaheuristics. *Opt. and Laser Technol.* 138, 106890. doi:10.1016/j.optlastec.2020.106890

Zhang, H., Duan, Z., Zheng, N., Li, Y., Zeng, Y., and Shi, W. (2022). An efficient class-constrained dbscan approach for large-scale point cloud clustering. *IEEE J. Sel. Top. Appl. Earth Observations Remote Sens.* 15, 7323–7332. doi:10.1109/JSTARS.2022.3201991

Zhang, J., Zhao, X., Chen, Z., and Lu, Z. (2019). A review of deep learningbased semantic segmentation for point cloud. *IEEE Access* 7, 179118–179133. doi:10.1109/ACCESS.2019.2958671

Zhao, H., Jiang, L., Jia, J., Torr, P. H., and Koltun, V. (2021). Point transformer. Proc. IEEE/CVF Int. Conf. Comput. Vis., 16259–16268. doi:10.48550/arXiv.2012.09164