remote sensing

MDPI

Article

Cross-Sensor Generalization of MLS Point-Cloud Semantic Segmentation

June Moh Goo [1,*], Zichao Zeng [1], Stefano Cavazzi [2], James Gillespie [2] and Jan Boehm [1]

[1] Department of Civil, Environmental and Geomatic Engineering, University College London, Gower Street, London WC1E 6BT, UK; zichao.zeng.21@ucl.ac.uk (Z.Z.); j.boehm@ucl.ac.uk (J.B.)
[2] Innovation and Research Department, Ordnance Survey, Explorer House, Adanac Park, Southampton SO16 0AS, UK; stefano.cavazzi@os.uk (S.C.); james.gillespie@os.uk (J.G.)
* Correspondence: june.goo.21@ucl.ac.uk

Highlights

What are the main findings?

- Segmentation accuracy on accumulated MLS point clouds falls from 0.91 to 0.67 mean IoU on the five shared classes when the test sensor is absent from the training data.
- Combining two labelled sensors, without any target data, raises mean IoU on the held-out third sensor by 0.081 over the average single source and reaches 97% of the score obtained by additionally self-training on unlabelled target data.

What are the implication of the main finding?

- Combining several labelled sensors, by joint training or by ensembling single-sensor models, is a strong target-free baseline for labelling point clouds from a new sensor, while self-training on unlabelled target data adds only 0.023 mean IoU.
- Without target labels, the best single source cannot be identified reliably (confidence-based selection picks the wrong source on one of three held-out sensors), so cross-sensor methods should be compared with the average over single sources and with label-free selection and ensembling, rather than with the best one.

Abstract

Semantic segmentation of mobile laser scanning (MLS) point clouds supports urban mapping, but models transfer poorly between acquisition systems. A new sensor brings a different beam geometry and intensity scale, and usually also a different platform, city, and annotation; we call this whole change a sensor domain shift. We study it with three labelled urban MLS datasets, Toronto3D, Paris-Lille-3D, and WHU-Urban3D, and an unlabelled 68-million-point Ordnance Survey (OS) point cloud of Southampton, using one pipeline, a unified label space, and Point Transformer V3. In-domain accuracy averages 0.91 mean intersection over union (mIoU) over the five shared classes and falls to 0.67 across sensors, mostly on pole and building. Under a leave-one-sensor-out protocol, combining the labelled sensors without target data raises held-out mIoU by 0.081 over the expected single-source score (0.079 ± 0.003 over three seeds) and reaches 97% of the score of a subsequent self-training step; joint training and an ensemble of single-source models perform on par. On three OS blocks labelled for evaluation, the three-source model reaches 0.945 mIoU against 0.739 for the average single source. Intensity standardized per cloud raises the cross-sensor mean unevenly, and normalized with source statistics it collapses from 8-bit to 16-bit data, so geometry remains the default input. The study provides a simple, deployable, target-free baseline for cross-sensor MLS segmentation.

Check for updates

Academic Editor: Sara Gonizzi Barsanti

Received: 26 October 2026
Revised: 20 November 2026
Accepted: 23 November 2026
Published: 28 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: LiDAR; mobile laser scanning; point cloud; semantic segmentation; domain generalization; cross-sensor transfer; multi-source training; urban mapping

1. Introduction

Mobile laser scanning (MLS) is a primary source of three-dimensional data for cities, and semantic segmentation turns its raw points into usable maps. Modern point-transformer backbones achieve high accuracy on public MLS benchmarks [1–3]. But these scores are measured in-domain: the same sensor, the same city, the same acquisition setup. Deployment is different. The point cloud a practitioner must label is usually unlabelled and comes from one specific sensor, while the only labelled data come from other sensors and other cities. The question that governs deployment is how well a model transfers from the sensors we have labels for to the sensor we must label.

National mapping agencies and cities maintain large 3D databases that must be kept current, and they resurvey with whatever sensor is available at the time [4,5]. Sensors and contractors change over time, so new data often comes from a different LiDAR than the labelled data [6]. A model that only works on its training sensor must be relabelled and retrained for every new unit, which is slow and expensive [7,8]. Label-efficient alternatives such as zero-shot detection with vision–language models reduce this cost, but so far address single classes such as buildings [9]. A model that transfers across sensors can label a new point cloud out of the box. Cross-sensor generalization is therefore not a one-off problem but a recurring need in operational mapping.

Transfer is hard because each LiDAR model samples the scene differently. A rotating 32-beam scanner draws objects as rings. The number and spacing of the rings follow the vertical field of view and the beam layout. The same pole or car side lands on different rings under a Velodyne, a Hesai, or a survey-grade scanner, and differently again when the same scanner is mounted at another angle. A network learns the sampling geometry of its training sensor, a kind of scan-pattern fingerprint, and loses accuracy on a new one [10–12]. We refer to this as a sensor domain shift. In practice a new sensor rarely arrives alone: the city, the vehicle and mounting, and the annotation campaign usually change with it. We therefore use *sensor domain* for the whole acquisition setting, the scanner together with its mounting, platform, city, and annotation, and every cross-sensor result in this paper should be read in this sense (Section 3.3).

Our case is an Ordnance Survey MLS point cloud of Southampton (OS point cloud). It has about 68 million points, no labels, and no usable colour, and it was collected with a Hesai PandarXT-32. To our knowledge, no public labelled MLS dataset uses this sensor. Models trained on a single source dataset behave very differently on this point cloud, and one of them misclassifies most buildings. Labelled MLS data is scarce, scan patterns differ across LiDAR sensors, and the city domains differ. With no labels on the target, the model cannot be fine-tuned either. Figure 1 shows that no single source transfers cleanly to the OS point cloud, and each produces different failure modes. This unlabelled real target motivates our work.

Unsupervised domain adaptation (UDA) uses the unlabelled target during training. A recent cross-sensor MLS method [13] aligns sensors through implicit surface reconstruction and feature discrepancy, but it needs the target data and a separate procedure for each target. Domain generalization (DG) uses no target data at all. COLA [14] showed that simply concatenating source domains is already a strong baseline. For an agency that must label a new sensor quickly, DG is more practical. If we combine several labelled source sensors and train one off-the-shelf model, how far can it go without any target labels? We

ask four questions. How much accuracy does a model lose when it moves from one LiDAR to another (Q1)? Which classes break down the most, and why (Q2)? Does training on combined source sensors improve generalization to an unseen sensor without target labels (Q3)? Which classes does the intensity channel help, and should it be used as input when the target sensor differs (Q4)?

The main contributions of this work are the following. *(i)* A controlled cross-sensor benchmark for accumulated MLS. It combines three labelled sensors (Toronto3D, Paris-Lille-3D, WHU-Urban3D) and one unlabelled operational target (Ordnance Survey, Hesai) in a single preprocessing pipeline and a unified 11-class label space with a shared-5 evaluation set. *(ii)* An evaluation protocol suited to an unlabelled target. It consists of a 3 × 3 transfer matrix, a leave-one-sensor-out protocol, and an average-single reference, the expected single-source score when target suitability is unknown, in place of an oracle choice of source. *(iii)* A target-free multi-sensor training baseline built on an off-the-shelf backbone, with two recipes (joint training and density augmentation) and a standard self-training reference that uses the target, applied end to end to the Ordnance Survey point cloud. *(iv)* Open code, configurations, and trained weights, together with a complete class-mapping table.

With this setup we find that the cross-sensor gap is large and concentrates on pole and building; the pole loss is consistent with scan-pattern sensitivity, while the building loss may equally reflect urban form and class balance. Target-free joint training raises held-out shared-5 mIoU by 0.081 over the average-single reference and reaches about 97% of the final self-training score. Ensembling the single-source models is an equally strong target-free alternative, whereas choosing one source by its confidence on the target is unreliable. Intensity helps reflectance-defined classes in domain, but across sensors its benefit is uneven and requires the target's own statistics, so we keep geometry as the main input (Section 5.3).

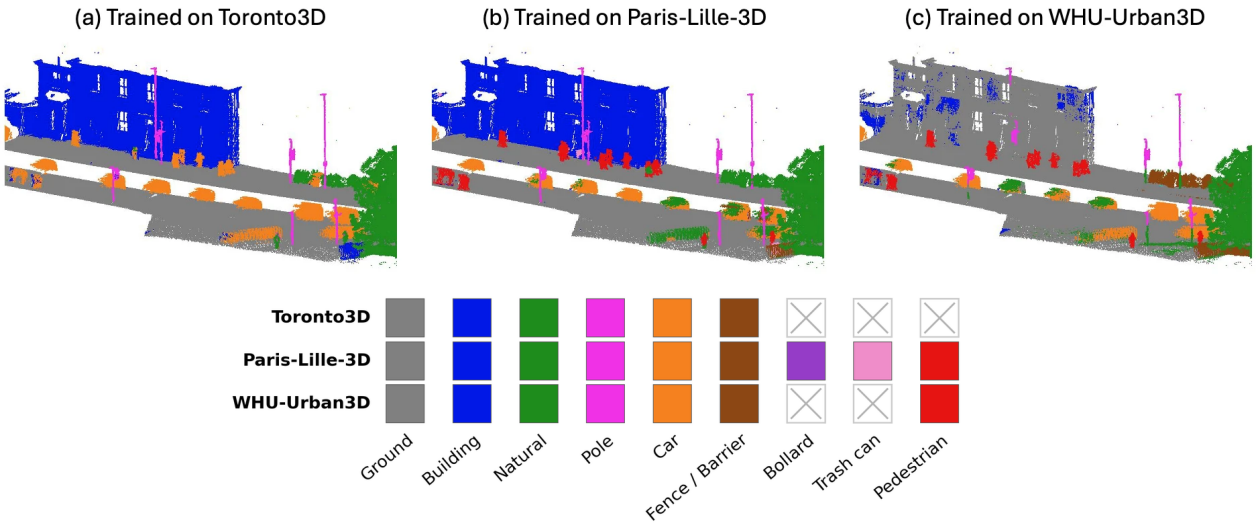

Figure 1. Single-source models on the unlabelled Ordnance Survey point cloud (Southampton, Hesai). Each is trained on one dataset and applied to OS without adaptation, coloured by the unified labels (legend below). In the legend, a cross means that the dataset has no label for that class. Road markings and utility line are not predicted in this crop and are omitted from the legend. The single-source models fail in different ways, and without target labels one cannot tell in advance which transfers best. (**a**) Toronto3D has no pedestrian label, so pedestrians are never predicted. (**b**) Paris-Lille-3D misses many cars and part of the ground (car recall 0.60). (**c**) The WHU-Urban3D model predicts most buildings as ground (building recall 0.13). WHU-Urban3D differs from OS in scanner and city and has the lowest building share of the sources. This motivates combining sources.

We frame this as a baseline study, not a new method. Pooling sources is not new in itself: COLA [14] showed its value for single-scan driving LiDAR. What is new is its evaluation on accumulated MLS against an unlabelled operational target, with a protocol that compares it with the expected score of a source chosen without target labels. Relative to COLA and later multi-dataset work [14,15], three things differ. First, the data are accumulated MLS rather than single sweeps, so the sensor signature is the arrangement of accumulated samples, not the sparsity of one scan. Second, the operational target has no training labels, so every method is compared with what a practitioner could achieve without target labels. Third, we measure how much of the score of a target-using self-training step the target-free model already reaches. It is a deployable reference for cross-sensor MLS, shown end to end on a real unlabelled national-mapping point cloud.

2. Related Work

2.1. Point Cloud Segmentation

Deep semantic segmentation of outdoor LiDAR has developed through three families of backbone. Point-based networks operate directly on coordinates. PointNet [16] and PointNet++ [17] introduced permutation-invariant pooling, and KPConv [18] added deformable point convolutions. Voxel and projection networks trade some geometric detail for efficiency. MinkowskiNet [19] and SparseConvNet [20] use sparse 3D convolutions. Range-image methods such as RangeNet++ [21] and cylindrical partitions such as Cylinder3D [22] exploit the scan geometry. Transformer networks apply attention over point neighbourhoods. Point Transformer and its successors [2,23,24] combine large receptive fields with high throughput and lead current outdoor benchmarks. We use Point Transformer V3 (PTv3) [2] through the Pointcept framework [25] as a fixed, off-the-shelf backbone. Our contribution is the cross-sensor study, not a new network. Most of these networks take coordinates as the core input. Colour and intensity are optional channels, and colour is often unavailable in MLS.

2.2. MLS for Urban Mapping

Mobile laser scanning is a key technology of urban remote sensing. Vehicle-mounted systems produce dense street-level point clouds. These feed road-asset inventories, high-definition maps, 3D city models, building and vegetation extraction, and the creation and updating of digital twins [26–29]. Semantic segmentation turns raw points into these products, so its reliability directly affects downstream mapping. The products must be maintained over time and across areas surveyed by different systems. The segmentation model must therefore work beyond the single sensor and city it was trained on. This operational reality frames our study.

2.3. Benchmark Datasets

Progress in outdoor point cloud segmentation has been driven by public benchmarks. Semantic3D [30] provides static terrestrial scans. SemanticKITTI [31] and nuScenes-lidarseg provide single-scan driving LiDAR [32,33]. SensatUrban [34] provides photogrammetric urban point clouds. For accumulated MLS, Toronto3D [26], Paris-Lille-3D [28], and the recent WHU-Urban3D [27] offer labelled street scenes from different cities and mapping systems. These datasets are almost always used one at a time, with in-domain train and test splits on the same sensor. We instead consider three of them as a single cross-sensor benchmark suite, together with an unlabelled operational point cloud. This lets us measure transfer between sensors rather than accuracy within one.

2.4. Domain Shift and UDA for LiDAR

A model trained on one LiDAR rarely transfers cleanly to another. The causes are well documented for driving LiDAR: different beam counts, mounting positions, ranges, point densities, and intensity statistics [35,36]. Two effects recur. First, the drop is class-dependent. Thin or sparsely sampled objects suffer most, because the number of beams that strike them changes sharply with beam geometry. Second, intensity is among the least transferable channels. It depends on each unit's emitter power, wavelength, gain, and range normalization [37]. Unsupervised domain adaptation (UDA) reduces the gap by using the unlabelled target during training. Common tools are confidence-filtered self-training, feature alignment, and geometric normalization that shrinks the sampling difference before features are compared [7,38,39]. For cross-sensor MLS, a recent approach [13] reconstructs a continuous implicit surface to place source and target in a common geometry and aligns features with maximum classifier discrepancy. These methods can be effective, but they need the target point cloud at training time and a separate procedure for each new target. We include a plain self-training step as our own UDA reference. The comparison then isolates the value of using target data at all, not the value of a particular design.

2.5. DG and Multi-Source Training

Domain generalization (DG) seeks models that transfer to unseen domains with no target data. The usual levers are augmentation, domain-invariant representation learning, and multi-source training. For LiDAR, augmentation-based DG has simulated beam dropout and density change to harden a single-source model [8,40]. Multi-source training by simple concatenation is a particularly practical recipe. COLA [14] relabels several driving datasets to a common coarse label set, concatenates them, and reports about a ten-point gain in domain generalization over single-source training. Later geometric-driven work confirms the value of this multi-dataset recipe [15]. This suggests that diversity in the training pool is itself a useful signal. Our joint multi-sensor training is a direct instance of this idea, applied to accumulated MLS across three labelled sensors and one unlabelled target sensor, and evaluated under a leave-one-sensor-out protocol.

2.6. Positioning

Our study sits in the DG setting and complements dedicated UDA methods. We do not propose a new adaptation network. We provide a controlled benchmark for accumulated MLS with three labelled sensors and one unlabelled operational target, a regime the mostly single-scan and single-dataset literature underserves. We characterize the cross-sensor gap per class and discuss its relation to beam geometry. We use an evaluation protocol suited to an unlabelled target, comparing against the average single source rather than an oracle. And we measure how much a simple target-free source combination captures against a standard self-training step. The result is a reproducible reference against which future cross-sensor MLS methods, including full UDA pipelines, can be measured.

3. Datasets and Sensors

We use three labelled source datasets and one unlabelled target. All four are accumulated urban MLS point clouds. The scene content is therefore broadly comparable, but the sources differ in more than the scanner. Table 1 lists the essentials: sensor, size, availability of colour and intensity, and the validation split. Each source also comes from a different city, platform, taxonomy, and annotation campaign, and the class balance differs sharply (Table 2). Table 3 lists these factors, and Section 6 discusses what our design can and cannot separate. Figure 2 shows an example crop from each dataset.

Table 1. The four datasets and their sensors. Each validation split is a whole tile or scene not seen in training. Points are the raw points in the files we use. For Paris-Lille-3D these are its four labelled files; the full release, including its unlabelled test areas, holds about 143 M points. The OS point cloud is unlabelled for training; three small blocks were labelled by Ordnance Survey for evaluation only (Section 5.8).

Dataset	Sensor	Points	RGB	Intensity	Classes	Validation
Toronto3D	Optech Maverick (Velodyne 32-ch, Teledyne Optech, Vaughan, ON, Canada)	78 M	yes	8-bit	8	L002
Paris-Lille-3D (NPM3D)	Velodyne HDL-32E (32-ch, Velodyne Lidar, San Jose, CA, USA)	120 M	no	8-bit	9	Lille2
WHU-Urban3D (Shanghai, MLS-S)	AS-900HL (survey-grade, CHC Navigation, Shanghai, China)	238 M	no	16-bit	18	6 scenes
OS Southampton (target)	Hesai PandarXT-32 (32-ch, Hesai Technology, Shanghai, China)	68 M	no	8-bit	none	3 labelled test blocks

Table 2. Share of labelled points per shared-5 class in each source (train and validation splits, unified label space, after voxelization). Class balance differs sharply between sources. WHU-Urban3D has few building points and is dominated by vegetation, and pole is rare everywhere but rarest in WHU-Urban3D.

Source	Ground	Building	Natural	Pole	Car
Toronto3D	45.5%	23.4%	19.1%	1.60%	6.54%
Paris-Lille-3D	41.4%	31.1%	19.2%	0.55%	3.22%
WHU-Urban3D	40.3%	4.63%	44.3%	0.12%	1.34%

Table 3. Factors of variation between the datasets. No pair of sources differs in the scanner alone. Density and spacing are from Section 3.3; class shares are from Table 2.

Factor	Toronto3D	Paris-Lille-3D	WHU-Urban3D	OS Southampton
Scanner head	Velodyne 32-ch (Optech Maverick)	Velodyne HDL-32E	AS-900HL (survey-grade)	Hesai PandarXT-32
Mounting/platform	vehicle-mounted (Optech Maverick system)	tilted 30°, rear of vehicle	vehicle-mounted	OS mapping vehicle
City, country	Toronto, Canada	Paris and Lille, France	Shanghai, China	Southampton, UK
Intensity	8-bit	8-bit	16-bit	8-bit
Raw density (pts/m^2)	742	1720	1091	993
Median NN spacing (cm)	1.4	1.5	1.5	2.2
Native classes	8	9	18	none
Unified classes annotated	8	9	9	none
Building/pole share	23.4%/1.60%	31.1%/0.55%	4.63%/0.12%	unknown
Annotation	point-wise semantic	point-wise semantic and instance	point-wise semantic and instance	none for training; 3 test blocks

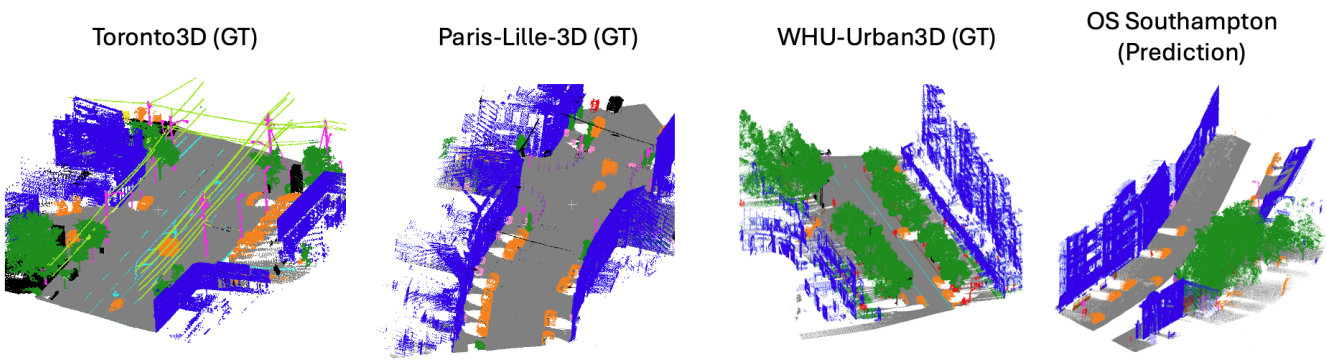

Figure 2. Example crops from the four datasets in an oblique view. The three sources are coloured by ground truth and OS by the three-source prediction with the unified labels. Colours: ground (grey), building (blue), natural (green), pole (magenta), car (orange), road marking (cyan), utility line (light green), fence/barrier (brown), bollard (purple), trash can (pink), pedestrian (red), and ignored points (black).

3.1. Labelled Source Datasets

The three sources are standard public labelled MLS datasets, and together they give diversity in sensor, city, and taxonomy. Toronto3D (Toronto, Teledyne Optech Maverick with a Velodyne 32-channel head) is the only one with colour, and it has eight classes. Paris-Lille-3D (Paris and Lille, Velodyne HDL-32E tilted by 30° at the rear of the vehicle [28]), also released as NPM3D, has nine classes and the same channels as our target, intensity but no colour, which makes it the closest match in input to OS. WHU-Urban3D (Shanghai, MLS-S partition, survey-grade AS-900HL) is the largest source and the most dissimilar, with 16-bit intensity and a fine 18-class taxonomy. The three taxonomies do not agree, with 8, 9, and 18 classes, so we remap them to a common label space in Section 4.2. Each validation split holds out whole tiles or scenes, so no training area overlaps it. Class imbalance is a well-known property of MLS point clouds, where ground, facades, and vegetation dominate and thin objects such as poles are rare [41,42]. Table 2 shows that the balance also differs sharply between sources.

3.2. The Unlabelled Target: OS Southampton

Our target is an Ordnance Survey MLS point cloud of Southampton, UK, acquired on 25 September 2025 (from the GPS timestamps in the data) with a Hesai PandarXT-32, a 32-channel spinning scanner, on an Ordnance Survey mobile mapping vehicle. It is delivered as two LAZ files, soton_l1 and soton_l2, of 41.9 and 26.6 million points, about 68 million in total, georeferenced in OSGB36/British National Grid (EPSG:27700) with heights in metres, and covering a block of about 0.7 km by 0.9 km. The cloud carries 8-bit intensity but no colour and no labels. It is a genuine deployment case: there is no ground truth for training, so the only option is to train on the labelled sources and transfer. Ordnance Survey additionally labelled three small blocks of soton_l2. They are used only to evaluate the final models (Section 5.8) and never for training or model selection. The OS cloud's channels match Paris-Lille-3D, intensity but no colour, which matters for the intensity question in Sections 5.3 and 5.8. The sensor was identified from the raw data packets. It is unlike any of the source units, which is why we expect a sensor domain shift on OS. Table 4 gives its sampling statistics next to the three sources.

Table 4. Raw sampling statistics of the four sensors, computed on all raw scenes or tiles of each dataset before preprocessing. Area is the footprint of occupied 1 m cells. Density is raw points per occupied square metre. NN spacing is the median distance to the nearest neighbouring point, with the interquartile range in parentheses, measured inside random 60 m windows with all points present. The last two columns describe the 5 cm voxel grid used for training: raw points per occupied voxel, and the share of raw points that survive the grid. WHU-Urban3D includes all 38 raw scenes here, whereas Table 1 counts the labelled ones.

Dataset	Area (10^3 m^2)	Density (pts/m^2)	NN Spacing (cm)	Points per 5 cm Voxel	Voxel Survival
Toronto3D	105.6	742	1.4 (0.4–2.4)	3.91	26%
Paris-Lille-3D	69.7	1720	1.5 (1.2–2.3)	2.63	38%
WHU-Urban3D	280.3	1091	1.5 (0.9–3.3)	2.73	37%
OS Southampton	69.0	993	2.2 (1.4–3.7)	2.08	48%

3.3. Sensor Domain

Throughout, *sensor* means the whole acquisition system: the scanner head, its mounting, and the platform, together with the city and the annotation campaign that come with each dataset, since in our data these change together. The units differ in beam count, vertical layout, mounting, and the density of a single scan. This should matter most for thin, structured objects. A pole, a pedestrian, or a car edge is defined by a small number of returns, and which returns land on it depends on the beam layout, so its appearance changes from one sensor to another. Extended classes such as road, ground, facades, and tree canopy cover large areas and are hit by many beams at once, so their local appearance does not hinge on where individual beams fall, and they should stay more stable across sensors. We therefore expect the cross-sensor gap to concentrate on thin, structured classes rather than extended ones, and Section 5.2 shows this holds for pole, though less clearly for building.

Table 4 quantifies the sampling of each unit on the raw point clouds, before any preprocessing. Once returns are accumulated along the drive, gross density differs by at most a factor of about 2.3 across the four units. The median spacing to the nearest neighbouring point is 1.4 to 2.2 cm, and the 5 cm voxel grid of Section 4.1 leaves 2.1 to 3.9 raw points per occupied voxel, so the grid removes most of the remaining difference in point count. What survives preprocessing is the arrangement of samples rather than their number: the ring spacing set by the vertical beam layout, the overlap of repeated passes, and the vertical coverage of facades. WHU-Urban3D shows the widest spread of spacings, with an interquartile range of 0.9 to 3.3 cm and near-duplicate returns in several scenes from overlapping passes, and OS is the sparsest unit at the 5 cm scale. These statistics make a simple density explanation unlikely for the gap measured in Section 5.2. They do not by themselves separate beam geometry from the non-sensor factors discussed in Section 6.

Table 3 shows that no pair of sources differs in the scanner alone. Toronto3D and Paris-Lille-3D are the closest pair, but they still differ in mounting and platform, city and country, native taxonomy, and class balance. WHU-Urban3D differs from both in almost every factor, including the urban form and a much lower building share. The OS target differs from all three sources in scanner, platform, and city. Every cross-sensor result below therefore measures a change of the whole acquisition setting, of which the scanner is one part.

4. Methods

Figure 3 summarizes the method. Its layout and visual styling were refined with a generative AI tool (Claude Opus 5.5, Anthropic); its content and structure were designed by the authors (see Acknowledgments).

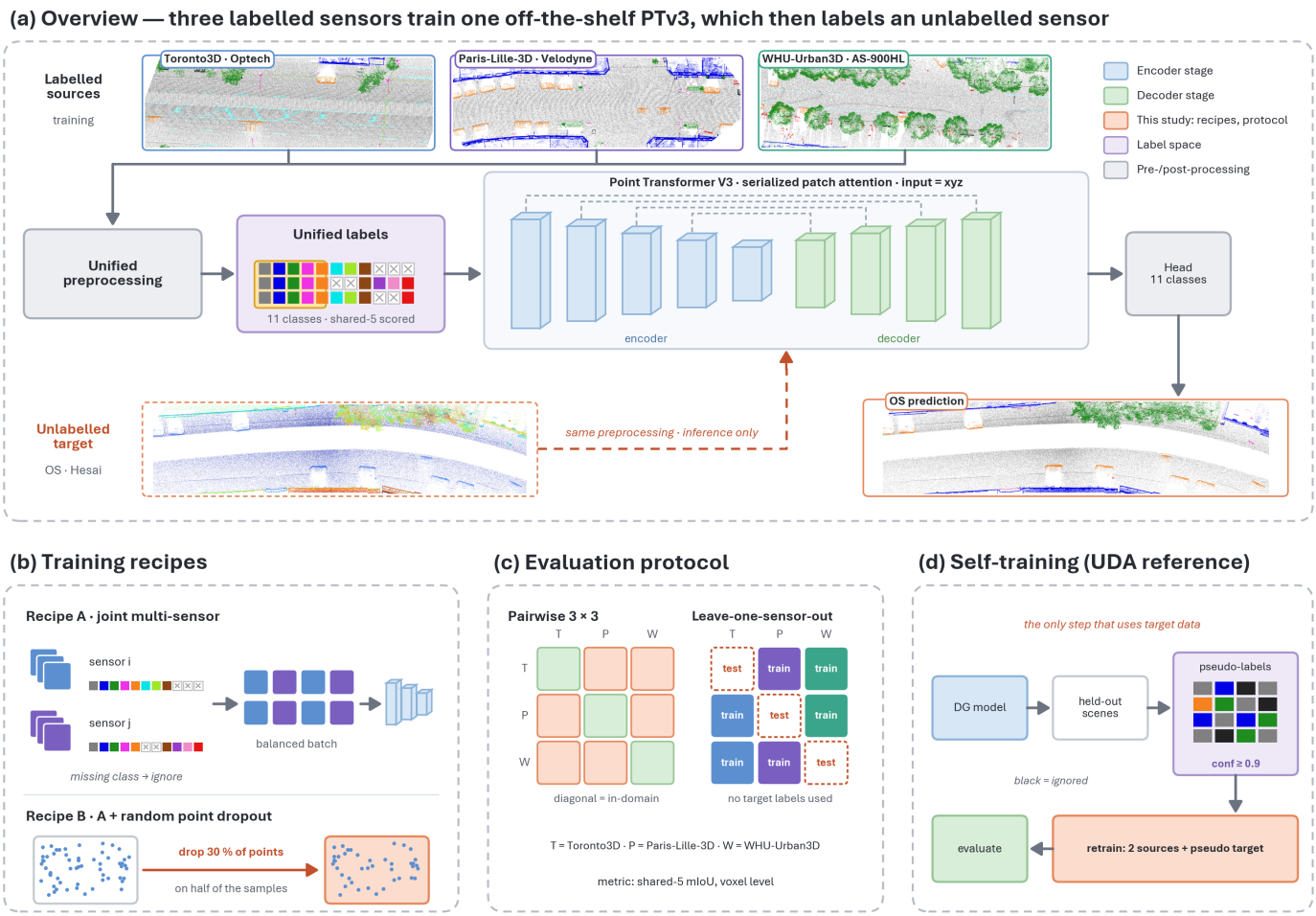

Figure 3. Overview of the method. (**a**) Three labelled sensors pass through one preprocessing pipeline and a unified label space and train an off-the-shelf Point Transformer V3 on coordinates only. The unlabelled OS point cloud follows the same preprocessing and is used for inference only. (**b**) Recipe A pools the labelled sensors, and Recipe B adds random point dropout. (**c**) Pairwise and leave-one-sensor-out evaluation, scored by shared-5 mIoU. (**d**) Self-training on the held-out sensor, the only step that uses target data. Pseudo-labels below 0.9 confidence are ignored.

4.1. Unified Preprocessing

Every point cloud, source and target alike, passes through the same pipeline. This prevents a cross-sensor effect from being an artefact of mismatched preprocessing. It also lets the unlabelled target be handled exactly as the sources. The pipeline has five steps. We downsample to a 5 cm voxel grid, keeping one real point per voxel. We remap native labels to the unified space of Section 4.2. We subtract a per-cloud global offset and store coordinates as float32. We standardize intensity per cloud to zero mean and unit variance. Finally, we cut each cloud into 50 m tiles, overlapping for training and disjoint for validation. Only the label remap uses labels, so the target passes through the whole pipeline unchanged. The intensity step concerns only the coord+int variant, and Section 5.3 discusses its use of target statistics. The main cross-sensor models use coordinates only and remain fully target-free. We do not re-weight or resample classes in the main experiments, and Section 5.7 tests class-weighted training.

4.2. Unified Label Space

The three sources do not share a class taxonomy. Toronto3D labels 8 classes, Paris-Lille-3D 9, and WHU-Urban3D 18. The same object often sits under a different name. To

measure transfer we map all of them into one 11-class space: ground, building, natural, pole, car, road marking, utility line, fence/barrier, bollard, trash can, and pedestrian. The mapping merges classes that mean the same thing. Toronto3D's fence and Paris-Lille-3D's barrier become one fence/barrier class. The finer surface classes and pole-like furniture of WHU-Urban3D collapse into the matching unified class, and its remaining street furniture is ignored (Appendix A). Figure 4 shows the space. Table A1 in Appendix A lists the complete mapping from every original class to its unified class or to ignore, with the rationale for the less obvious merges.

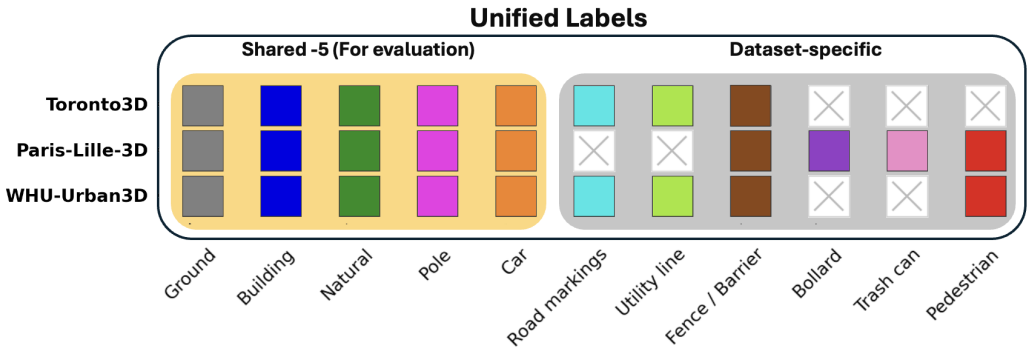

Figure 4. The unified 11-class label space. Rows are the three labelled sources, columns are the classes. A coloured square means the source annotates that class. A cross means those points are set to an ignore label. The five left classes (Ground, Building, Natural, Pole, Car) are annotated by all three sources with consistent definitions and form shared-5, which carries every cross-sensor score. Colours are consistent across all figures.

No source annotates all 11 classes. When a class is missing from a source, its points are set to an ignore label, not forced into a wrong class. Toronto3D has no pedestrian category, so a person on a Toronto3D street is ignored during training rather than counted as pole or building. The network keeps all 11 outputs. Only the supervision is partial per source. A jointly trained model can therefore predict a class on the target that one of its sources never labelled.

Six classes are annotated by all three sources. Fence/barrier is one of them, but its native definitions differ (fence; barrier; fence and road divider), so we keep the five consistently defined classes, ground, building, natural, pole, and car, as shared-5. We call these shared-5 and mark them in Figure 4. Every cross-sensor number in this paper is computed on shared-5. Each transfer direction is then scored on the same label set, and the comparison stays fair. The other five classes are annotated by only one or two sources. A cross-sensor score on them would compare different label sets, so we keep them out of the headline metric. They still take part in training, and we report them in-domain where a single source allows it.

We use the corrected release v1.1 of WHU-Urban3D, in which the maintainers relabelled one building in validation scene 4629 that was annotated as pedestrian in the original release. All WHU-Urban3D results in this paper are scored with the v1.1 labels; Appendix B reports the effect of the correction and a check of the small-object labels of all sources.

4.3. Model and Training

We use Point Transformer V3 as an off-the-shelf backbone, unchanged, through the Pointcept framework [2,25]. Training uses cross-entropy plus Lovász softmax loss [43], AdamW (learning rate 0.002, 0.0002 for the transformer blocks, weight decay 0.005) with a one-cycle schedule, a batch size of 8, 100 epochs, and two NVIDIA RTX 3090 GPUs. The configurations follow the Pointcept PTv3 outdoor setting at commit d727225 and are released with the code. We train two input variants: coordinates only (*coord*) and coordinates plus

standardized intensity (*coord+int*). Standard augmentations are always used: rotation, scaling, flip, jitter, Mix3D-style scene mixing [44], and a random sphere crop of up to 102,400 points.

The experiments compare two training recipes. Recipe A is joint training on several source sensors in the unified label space. Recipe B adds random point dropout on top of Recipe A, to mimic the density and beam differences between sensors. Half of the training samples, chosen at random, lose 30% of their points before voxelization. It is a simple form of the density augmentation used for single-source LiDAR domain generalization [8]. Training runs in fp16. Inference runs in fp32, because the fp16 sparse-convolution tuner is unstable at test time. For the unlabelled target, voxel predictions are propagated to the full-resolution point cloud by nearest neighbour.

4.4. Protocol and Baselines

All cross-sensor numbers are shared-5 mean IoU at the voxel level, without test-time augmentation (TTA). Every comparison is therefore on equal footing. We run four experiments.

- In-domain: Train and test on the same sensor, for both input variants (Section 5.1).
- Pairwise cross-sensor: Apply each single-sensor model to every sensor, giving a 3×3 matrix (Section 5.2).
- Leave-one-sensor-out (LOSO): Hold out one sensor, train jointly on the other two, and test on the held-out sensor (Section 5.4). Model selection uses only the training sensors' validation splits, so the held-out sensor never leaks. Recipes A and B are compared with an equal number of chunks per sensor, ruling out a size imbalance.
- UDA self-training: The joint model pseudo-labels the held-out sensor's unlabelled training scenes, keeping points above 0.9 softmax confidence. The model is then retrained on the two labelled sources plus the pseudo-labelled target (Section 5.5). This is the only experiment that uses target data.

For an unlabelled target, one cannot tell in advance which single source transfers best. That choice would itself need target labels. We therefore compare joint training against the *average-single* reference, the mean score of the available single-source models. It is not a deployable model but the expected single-source score when target suitability is unknown, that is, when a source is picked without any target-based criterion. We also report the best single source as a reference. It is, however, an upper bound that a practitioner could not select without labels.

5. Results

5.1. In-Domain and the Intensity Finding (Q4)

Two metrics appear in this paper. Shared-5 mIoU, the mean IoU over the five classes common to all sources, carries every cross-sensor comparison. The in-domain mIoU in Table 5 is instead over each source's own class set, and it serves only to show the effect of intensity. In domain, shared-5 mIoU is the diagonal of Table 6. Table 5 reports in-domain per-class IoU for the two input variants. The largest effect is the same on the two sources that annotate road marking: intensity helps reflectance-defined classes. Road marking is flat and defined by high reflectance. With intensity it jumps from near zero to usable (Toronto3D 0.001 to 0.564; WHU-Urban3D 0.104 to 0.446). On the shared-5 classes its effect is mixed, and large in one case. It raises Paris-Lille-3D pole (0.777 to 0.842) and WHU-Urban3D natural (0.880 to 0.920) but lowers WHU-Urban3D building (0.827 to 0.658) and pole (0.853 to 0.820). Pedestrian, a purely geometric class, is better without it (Paris-Lille-3D 0.950 vs. 0.909). This answers Q4 in domain. Section 5.3 tests whether intensity also helps across sensors.

Table 5. In-domain per-class IoU on the shared-5 classes, coord vs. coord+int, at the voxel level without TTA. The mIoU column is over each source's own annotated classes (Toronto3D 8, Paris-Lille-3D 9, and for WHU-Urban3D the 9 unified classes that occur in its validation scenes, evaluated on all six validation scenes with the v1.1 labels), so it is not comparable across rows. The diagonal of Table 6 scores the same coord models on shared-5 only and ignores points labelled outside shared-5, so it is slightly higher than the mean of the five columns here. Road marking (not a column) is where intensity helps: Toronto3D 0.001 → 0.564, WHU-Urban3D 0.104 → 0.446. All rows are scored with the corrected v1.1 labels in one seeded evaluation pass. The coord+int models are those of the per-cloud condition in Section 5.3, trained with the configuration of Section 4.3.

Dataset	Variant	Ground	Building	Natural	Pole	Car	mIoU (Own Classes)
Toronto3D	coord	0.947	0.920	0.960	0.786	0.911	0.715
	coord+int	0.958	0.910	0.961	0.765	0.931	0.791
Paris-Lille-3D	coord	0.979	0.983	0.883	0.777	0.978	0.862
	coord+int	0.973	0.980	0.872	0.842	0.982	0.877
WHU-Urban3D	coord	0.862	0.827	0.880	0.853	0.929	0.689
	coord+int	0.862	0.658	0.920	0.820	0.920	0.707

Table 6. Cross-sensor transfer, shared-5 mIoU (coord, voxel, no TTA). Diagonal (bold) is in-domain.

Train/Test	Toronto3D	Paris-Lille-3D	WHU-Urban3D
Toronto3D	**0.916**	0.854	0.653
Paris-Lille-3D	0.680	**0.927**	0.613
WHU-Urban3D	0.636	0.573	**0.886**

5.2. The Cross-Sensor Gap (Q1, Q2)

With the coord model we measure transfer across sensors. Table 6 gives the 3 × 3 matrix and Figure 5 shows it as a heatmap. The diagonal, in-domain shared-5 mIoU, ranges from 0.886 to 0.927 (mean 0.91). Off the diagonal it falls to 0.573 to 0.854, a mean drop of about 0.24 mIoU. This answers Q1: the gap is large. WHU-Urban3D is the hardest target and the weakest source, consistent with its survey-grade sensor and geographic difference.

The per-class breakdown answers Q2. The loss is not spread evenly. Pole falls to 0.18 when a Paris-Lille-3D or WHU-Urban3D model is applied to Toronto3D. Building falls to 0.33 when a Toronto3D or Paris-Lille-3D model is applied to WHU-Urban3D, and to 0.30 when the WHU-Urban3D model is applied to Paris-Lille-3D. Ground and natural stay high everywhere, at 0.63 to 0.95 and 0.73 to 0.91. These are the extended classes of Section 3.3, and this holds for a tree canopy as much as for a flat road, even though only the road is geometrically smooth. Thin, compact objects such as poles suffer most, as anticipated there. Building is also extended but still drops on the WHU-Urban3D pairs, which we relate to urban form and class balance in Section 6.

Class frequency is a second factor. Ground and natural are also the dominant classes, and pole is rare in every source (Table 2). Frequency alone, however, does not explain the drop. Within one sensor rare classes are learned well. Pole reaches 0.78 to 0.85 IoU in domain (Table 5), even in WHU-Urban3D where it makes up only 0.12% of the points. Car is also a minority class, yet it keeps 0.57 to 0.96 across sensors, whereas pole falls to 0.18 to 0.62. The loss therefore appears when the acquisition setting changes rather than from rarity alone, although our data cannot separate the two effects completely.

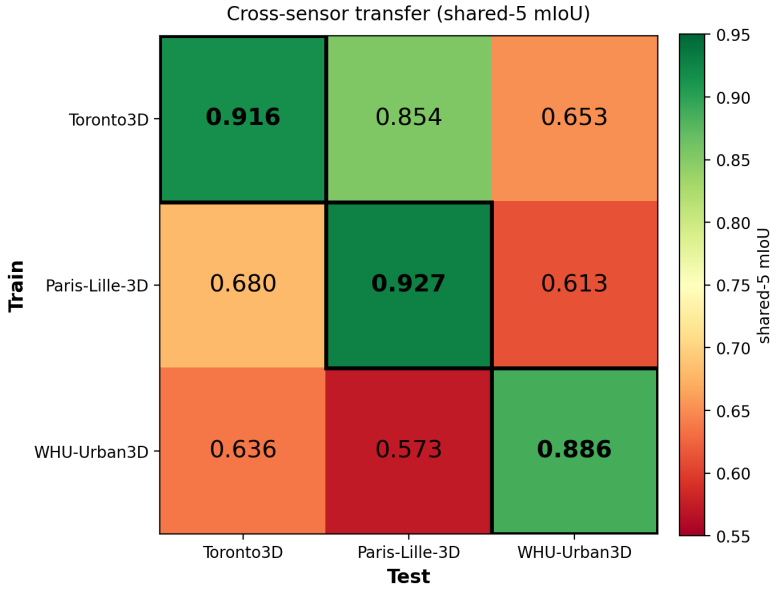

Figure 5. Cross-sensor transfer matrix as a heatmap (shared-5 mIoU). The diagonal is in-domain (mean 0.91). The off-diagonal cross-sensor mean is 0.67, a drop of about 0.24.

5.3. Intensity Across Sensors (Q4)

Section 5.1 answered Q4 only in domain. We now test intensity across sensors directly. We train coord+int single-source models under two normalizations and score the full 3 × 3 matrix (Table 7; per-class IoU in Appendix C). *Per-cloud* standardization is the one of Section 4.1: every cloud, the target included, is standardized with its own statistics, so it uses the target's intensity distribution but no target labels. *Source-only* standardization is fully inductive: the raw intensity of every cloud is standardized with the mean and standard deviation of the source's training data, and the target's statistics are never used. All models share the configuration of the coord models, and each is a single training run.

Table 7. Intensity across sensors, shared-5 mIoU (voxel, no TTA) for three input conditions. Per-cloud: intensity standardized with each cloud's own statistics (transductive, no target labels). Source-only: standardized with the source's training statistics (fully inductive). All values come from one seeded evaluation pass with WHU-Urban3D v1.1 labels. The coord row is re-scored in this pass and differs from Table 6 by at most 0.005 because of the random voxel selection at test time. Single training run per model. Best per row in bold.

Source → Target	Coord	Coord+Int (per-Cloud)	Coord+Int (Source-Only)
Toronto3D → Paris-Lille-3D	0.857	**0.880**	0.853
Toronto3D → WHU-Urban3D	**0.658**	0.643	0.001
Paris-Lille-3D → Toronto3D	0.677	**0.727**	0.675
Paris-Lille-3D → WHU-Urban3D	**0.611**	0.573	0.001
WHU-Urban3D → Toronto3D	0.634	0.709	**0.712**
WHU-Urban3D → Paris-Lille-3D	0.578	**0.743**	0.551
Cross-sensor mean	0.669	**0.712**	0.466
In-domain mean (diagonal)	0.910	0.902	**0.916**

Per-cloud intensity raises the cross-sensor mean from 0.669 to 0.712, but the gain is uneven. It is largest with WHU-Urban3D as the source, towards Paris-Lille-3D (0.578 to 0.743, mostly building, 0.31 to 0.87) and towards Toronto3D (0.634 to 0.709). It also helps between Toronto3D and Paris-Lille-3D, for example Paris-Lille-3D pole on Toronto3D (0.17 to 0.31). It lowers both transfers to WHU-Urban3D, the hardest target, from Toronto3D (0.658 to 0.643) and from Paris-Lille-3D (0.611 to 0.573), and in domain it lowers WHU-

Urban3D building (Table 5). Source-only standardization keeps in-domain accuracy and transfers between the two 8-bit sources, but it collapses to 0.001 when a Toronto3D or Paris-Lille-3D model meets the 16-bit WHU-Urban3D intensity, whose values then lie hundreds of standard deviations outside the training range. In the reverse direction, 8-bit values map to an almost constant value under the WHU-Urban3D statistics. The WHU-Urban3D model then effectively loses the channel, yet it scores 0.712 and 0.551 against 0.634 and 0.578 for coord. Since this model gains almost no usable intensity, differences of this size between single training runs cannot be attributed to intensity alone, and the per-pair intensity effects in Table 7 should be read with this run-to-run spread in mind; only the per-cloud gain on WHU-Urban3D → Paris-Lille-3D (0.578 to 0.743) clearly exceeds it.

This completes the answer to Q4. Intensity can carry information across sensors, but here only when each cloud, the target included, was standardized with its own statistics. Even then the effect depends on the sensor pair and is negative on the hardest target, for single-source and joint models alike (Table 8), and an unlabelled target offers no way to check which case applies. We therefore keep coordinates for the main experiments, which stay fully target-free, and treat per-cloud intensity as an option to validate on a small labelled target sample.

Table 8. Joint LOSO training (Recipe B) with and without per-cloud intensity, shared-5 mIoU on the held-out sensor, in one seeded evaluation pass with WHU-Urban3D v1.1 labels. Avg single uses the single-source models of the same input condition (Table 7). Single training run for coord+int. Best per row in bold.

Held-Out	Avg Single (Coord)	DG (B), Coord	Avg Single (Coord+Int)	DG (B), Coord+Int
Toronto3D	0.656	0.703	0.718	**0.729**
Paris-Lille-3D	0.718	0.843	0.812	**0.888**
WHU-Urban3D	0.635	**0.704**	0.608	0.642
Mean	0.669	0.750	0.713	**0.753**

We also retrained the joint LOSO model (Recipe B) with per-cloud intensity (Table 8). Its held-out mean is on par with that of the coord model (0.753 vs. 0.750), but the two differ by sensor. Intensity helps on Paris-Lille-3D (0.843 to 0.888, mostly pole) and on Toronto3D (0.703 to 0.729), and it hurts on WHU-Urban3D (0.704 to 0.642, mostly pole, 0.63 to 0.39; per-class values from the same evaluation pass as Table 8). The Paris-Lille-3D and WHU-Urban3D changes exceed the seed spread of the coord models (Section 5.5), but given the run-to-run spread noted above and the different hardware setup (see below), all per-sensor differences are indicative only. Because intensity already lifts the single-source models, joint training with intensity gains 0.040 over its own average-single reference (0.713), against 0.081 for coord. This model was trained on one GPU with a batch of 4 and two-step gradient accumulation, giving the same effective batch of 8 (normalization statistics, however, are computed over 4 samples per step), and it is a single run.

5.4. Joint Multi-Sensor Training (Q3)

Table 9 gives the LOSO per-class results and Figure 6 summarizes them. For each held-out sensor we list the two single sources, their average (the expected single-source score), and joint training with Recipes A and B. Against the average-single reference, Recipe B wins on every held-out sensor, by 0.081 mean shared-5 mIoU (0.668 to 0.749) and by up to 0.122 on Paris-Lille-3D. Over three training seeds the gain is 0.079 ± 0.003. This answers Q3. Pooling source sensors, with no target labels, clearly beats picking a single source blindly. Density augmentation adds little over plain joint training (Section 5.7). The main lever is combining sources; our design does not separate their diversity from the larger total amount of labelled data.

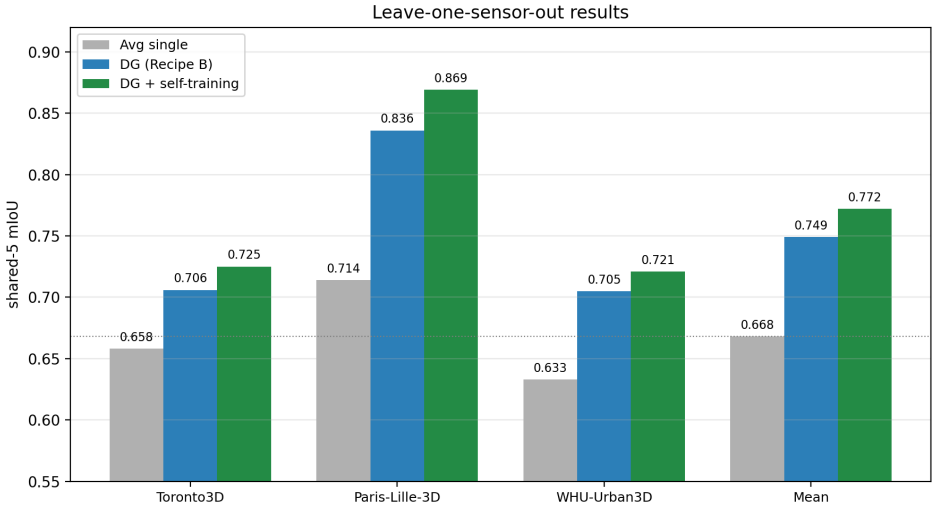

Figure 6. LOSO summary. Target-free joint training (DG, Recipe B) beats the average-single reference on every held-out sensor, and self-training adds a small further gain. Dotted line: mean average-single reference.

Table 9. LOSO per-class shared-5 IoU. For each held-out target: single sources, their average, and joint Recipes A/B (balanced). * Retrained with an identical configuration (Appendix B). Gr = ground, Bu = building, Na = natural, Po = pole, Ca = car. Bold: Recipe B, used in the rest of the paper.

Held-Out	Training	Gr	Bu	Na	Po	Ca	mIoU (Shared-5)
Toronto3D	Paris-Lille-3D only	0.95	0.82	0.88	0.18	0.57	0.680
	WHU-Urban3D only	0.88	0.63	0.80	0.18	0.70	0.636
	avg single (reference)						0.658
	Paris-Lille-3D+WHU-Urban3D (A)	0.95	0.85	0.87	0.15	0.60	0.683
	Paris-Lille-3D+WHU-Urban3D (B)	0.96	0.86	0.88	0.17	0.66	**0.706**
Paris-Lille-3D	Toronto3D only	0.95	0.95	0.80	0.61	0.96	0.854
	WHU-Urban3D only	0.63	0.30	0.73	0.38	0.82	0.573
	avg single (reference)						0.714
	Toronto3D+WHU-Urban3D (A)	0.95	0.96	0.79	0.54	0.95	0.838
	Toronto3D+WHU-Urban3D (B)	0.95	0.95	0.74	0.58	0.97	**0.836**
WHU-Urban3D	Toronto3D only	0.80	0.33	0.88	0.62	0.65	0.653
	Paris-Lille-3D only	0.80	0.33	0.91	0.44	0.59	0.613
	avg single (reference)						0.633
	Toronto3D+Paris-Lille-3D (A)	0.80	0.33	0.91	0.69	0.76	0.698 *
	Toronto3D+Paris-Lille-3D (B)	0.81	0.35	0.91	0.64	0.82	**0.705**

5.5. Self-Training (UDA)

So far no target data was used. We now add one standard self-training step on the unlabelled target (Section 4.4). Table 10 summarizes the held-out means over the three labelled sensors; the fourth target, OS, has no training labels and is evaluated on three labelled blocks in Section 5.8. Self-training lifts the mean by a further 0.023, to 0.772 (0.022 ± 0.003 over three training seeds). The important reading is the ratio. Target-free joint training already reaches about 97% of the final score obtained with this self-training procedure (0.749 of 0.772). This is a ratio of final scores, not of adaptation gains. Of the 0.104 total improvement over the average-single reference, joint training contributes 0.081 and self-training the remaining 0.023 (rounded), so most of the gain that this self-training step achieves is captured without touching the target. Table 10 also lists the best single source for reference. It is an upper bound that cannot be selected without target labels. DG exceeds it on the mean (0.749 vs. 0.729) and on two of the three sensors, falling short only

on Paris-Lille-3D, where the three-seed mean of 0.852 ± 0.016 is within seed variation of the oracle's 0.854, and DG plus self-training exceeds it on every sensor.

Table 10. Held-out means, shared-5 mIoU. "avg single" is the expected single-source score when target suitability is unknown, not a deployable model. "best single" is a reference upper bound that needs target labels to select. The last row gives the mean ± standard deviation over three training seeds (the run reported above and two replicates with fixed seeds). Per held-out sensor (Toronto3D, Paris-Lille-3D, WHU-Urban3D), the three-seed means are 0.706 ± 0.010, 0.852 ± 0.016, and 0.683 ± 0.022 for DG (B), and 0.722 ± 0.004, 0.879 ± 0.012, and 0.705 ± 0.014 for DG + self-training. WHU-Urban3D is scored with the corrected v1.1 labels; its two replicates were retrained with an identical configuration (Appendix B).

Held-Out	Avg Single	DG (B)	DG + Self-Train	Best Single (Ref.)
Toronto3D	0.658	0.706	**0.725**	0.680
Paris-Lille-3D	0.714	0.836	**0.869**	0.854
WHU-Urban3D	0.633	0.705	**0.721**	0.653
Mean	0.668	0.749	**0.772**	0.729
Mean, 3 seeds	0.668	0.747 ± 0.003	**0.769 ± 0.005**	0.729

Bold indicates the best value in each row.

Figure 7 shows the LOSO results qualitatively. A single source fails, for example predicting a building as ground on Paris-Lille-3D. Joint training recovers the scene. Self-training barely changes it. Two class and sensor pairs resist every coordinate-only method, Toronto3D pole (about 0.15 to 0.18) and WHU-Urban3D building (about 0.35). Class-weighted training lifts Toronto3D pole only to 0.27 (Section 5.7). Source combination does not remove this limit. Whether it is sensor-intrinsic or tied to city and class balance is discussed in Section 6.

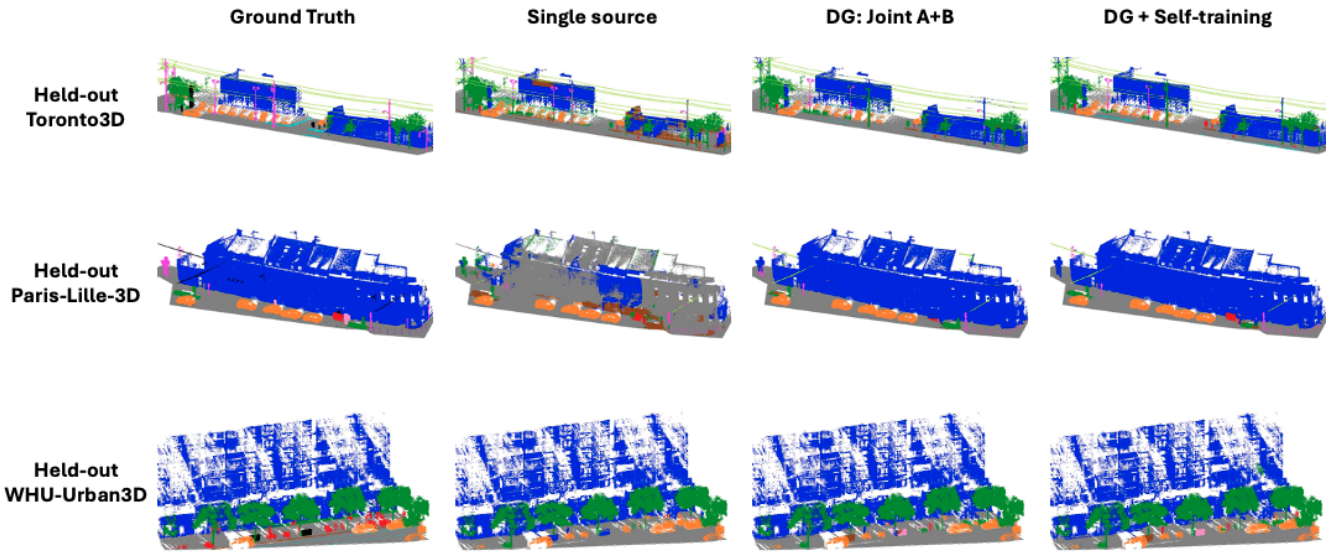

Figure 7. Qualitative LOSO results on the three held-out sensors. A single source fails, for example predicting a building as ground on Paris-Lille-3D. Joint training recovers the scene. Self-training barely changes it. The single-source column shows the WHU-Urban3D model for the Toronto3D and Paris-Lille-3D targets and the Toronto3D model for the WHU-Urban3D target. Colours follow the unified labels, as in Figure 2.

5.6. Ensembling and Source Selection Without Target Labels

Joint training is one way to use several sources without target labels. Two others need only the single-source models and, for selection, their predictions on the unlabelled target. An *ensemble* averages the softmax outputs of the two single-source models [45], each first

pooled into the shared-5 classes plus an "other" class. *Source selection* picks one model from its behaviour on the unlabelled target, either the higher mean maximum softmax probability or the lower mean normalized entropy. Agreement between models, a third common criterion, is symmetric when there are only two candidates and cannot choose between them, so we do not use it. Table 11 scores both together with the DG (B) model in one seeded evaluation pass.

Table 11. Target-label-free ways of using two source sensors, shared-5 mIoU on the held-out sensor. Selection picks the single-source model with the higher mean maximum softmax on the unlabelled target (lower mean entropy gives the same choice); the chosen source is in parentheses. Ensemble averages the two single-source softmax outputs pooled into shared-5. All columns come from one seeded evaluation pass with WHU-Urban3D v1.1 labels, so values shared with Table 10 differ by up to 0.007, and single-source values differ from Tables 7 and 8 by up to 0.005, because of the random voxel selection at test time. T = Toronto3D, P = Paris-Lille-3D.

Held-Out	Avg Single	Selection	Ensemble	DG (B)	Best Single (Ref.)
Toronto3D	0.657	0.677 (P)	**0.720**	0.703	0.677
Paris-Lille-3D	0.713	**0.853** (T)	**0.853**	0.843	**0.853**
WHU-Urban3D	0.636	0.613 (P)	0.672	**0.704**	0.658
Mean	0.669	0.714	0.748	**0.750**	0.729

Bold indicates the best value in each row.

Confidence and entropy pick the same source for every target. They pick the better source for Toronto3D and Paris-Lille-3D but the worse one for WHU-Urban3D, where the Paris-Lille-3D model is the more confident (mean maximum softmax 0.971 vs. 0.943) yet the less accurate (0.613 vs. 0.658). Selection reaches 0.714 on the mean, between the average-single reference (0.669) and the oracle best single (0.729). Under a sensor shift, confidence is therefore not a reliable guide to which source transfers best, in line with the known miscalibration of neural networks under dataset shift [46,47], which supports comparing against the average single source (Section 4.4).

The ensemble is much stronger. It matches joint training on the mean (0.748 vs. 0.750), is better on Toronto3D (0.720 vs. 0.703) and Paris-Lille-3D (0.853 vs. 0.843), and is worse on WHU-Urban3D (0.672 vs. 0.704). Most of the benefit of combining sensors can therefore be obtained at the output level as well as by joint training. The gain comes from combining sensors, whichever way they are combined, rather than from joint training as such. Joint training keeps practical advantages: one model at inference instead of one per source, and a single model that pseudo-labels the target for self-training.

5.7. Ablation

Table 12 and Figure 8 ablate the recipe. Adding density augmentation (A to B) changes the mean from 0.740 to 0.749. The gain is small and not consistent across sensors: +0.023 on Toronto3D, +0.007 on WHU-Urban3D, and −0.002 on Paris-Lille-3D. These per-sensor gains are of the order of the seed-to-seed standard deviation on the individual held-out sensors (0.010 to 0.022, Table 10). WHU-Urban3D is much larger than the other sources, so we also check that its volume is not the cause. Removing the per-sensor chunk balancing gives 0.752 against 0.740 balanced. The difference is mixed across folds: +0.029 on Toronto3D, −0.017 on Paris-Lille-3D, and +0.024 on WHU-Urban3D, where both Recipe A models were retrained. These per-fold differences are of the same order, so we do not read them as a consistent effect of balancing. All three recipes therefore lie within 0.012 of each other and well above the average-single reference (0.668). Class weighting is a common remedy for class imbalance [48,49]. We retrained Recipe B with class-weighted cross-entropy (Table 13). The held-out mean is unchanged (0.747 vs. 0.750). Weighting raises Toronto3D pole from 0.16 to 0.27 but lowers Paris-Lille-3D pole and WHU-Urban3D car, and WHU-Urban3D

building stays at 0.35. Weighting thus shifts errors between classes rather than reducing them. The weights follow the class balance of the training sources, which differs from that of each held-out sensor (Table 2). The dominant lever is combining sources, not balancing, augmentation, or class weighting. We keep Recipe B, the configuration used for the seed replicates and self-training.

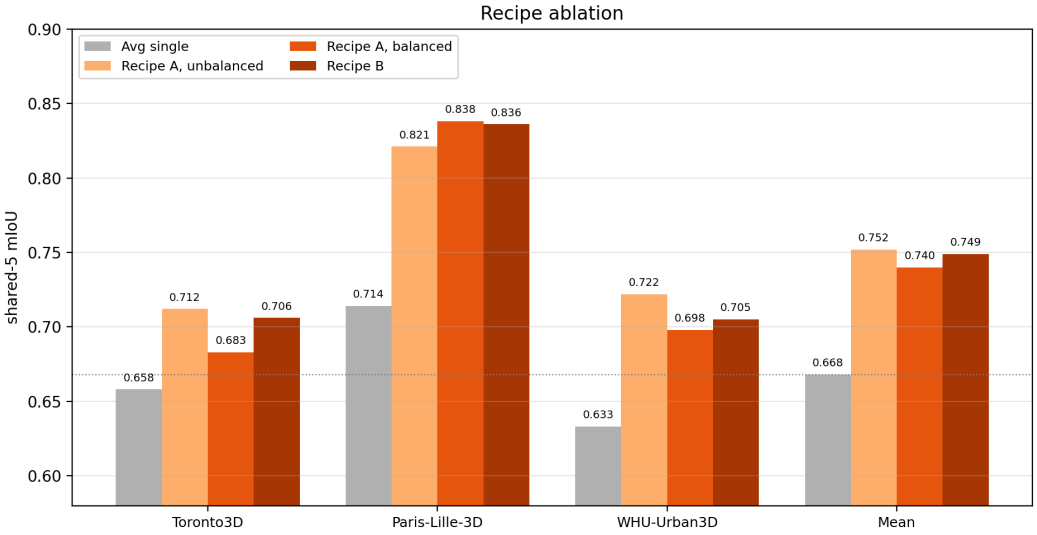

Figure 8. Ablation on LOSO mean shared-5 mIoU. All joint recipes beat the average-single reference. The three joint recipes lie within 0.012 of each other, small relative to their gain over the average-single reference, so neither chunk balancing nor density augmentation (Recipe B) is a consistent lever. Dotted line: mean average-single reference.

Table 12. Ablation on LOSO mean shared-5 mIoU. * Retrained with an identical configuration (Appendix B).

Recipe	Toronto3D	Paris-Lille-3D	WHU-Urban3D	Mean
A (joint, unbalanced)	0.712	0.821	0.722 *	0.752
A (joint, balanced)	0.683	0.838	0.698 *	0.740
B (joint + density aug)	0.706	0.836	0.705	0.749

Table 13. Joint LOSO training (Recipe B) with and without class-weighted cross-entropy, shared-5 IoU on the held-out sensor, in one seeded evaluation pass with WHU-Urban3D v1.1 labels. Weights are the inverse square root of class frequency in the training sources, capped at 10. Single training run for the weighted models.

Held-Out	Loss	Ground	Building	Natural	Pole	Car	mIoU
Toronto3D	Unweighted	0.963	0.855	0.884	0.160	0.652	0.703
	Weighted	0.922	0.841	0.900	0.268	0.768	0.740
Paris-Lille-3D	Unweighted	0.945	0.953	0.744	0.598	0.972	0.843
	Weighted	0.955	0.962	0.799	0.520	0.977	0.842
WHU-Urban3D	Unweighted	0.806	0.345	0.914	0.629	0.824	0.704
	Weighted	0.791	0.352	0.903	0.613	0.628	0.657
Mean	Unweighted						0.750
	Weighted						0.747

5.8. Application: Ordnance Survey Southampton

Finally we apply the three-source model (all sources, Recipe B, coord) to the unlabelled OS point cloud through the same pipeline. The models were trained and applied without any OS labels. Ordnance Survey additionally labelled three blocks of soton_l2, which we use for evaluation only (Table 14). We first describe the qualitative reading of the whole cloud. Figure 9 sets the three single-source models beside the three-source model on one crop. The single-source failures are the ones from Figure 1, and the three-source model resolves them. Over the whole cloud, which is unlabelled except for three blocks, class shares give a coarse check, and they show the largest failure being removed. The WHU-Urban3D-only model labels 75.9% of soton_l1 as ground and only 4.0% as building. The three-source model labels 46.8% as ground and 29.5% as building, close to the Toronto3D-only and Paris-Lille-3D-only models (32.1% and 32.9% building). Class shares are not accuracy, so this shows only that the gross confusion is corrected.

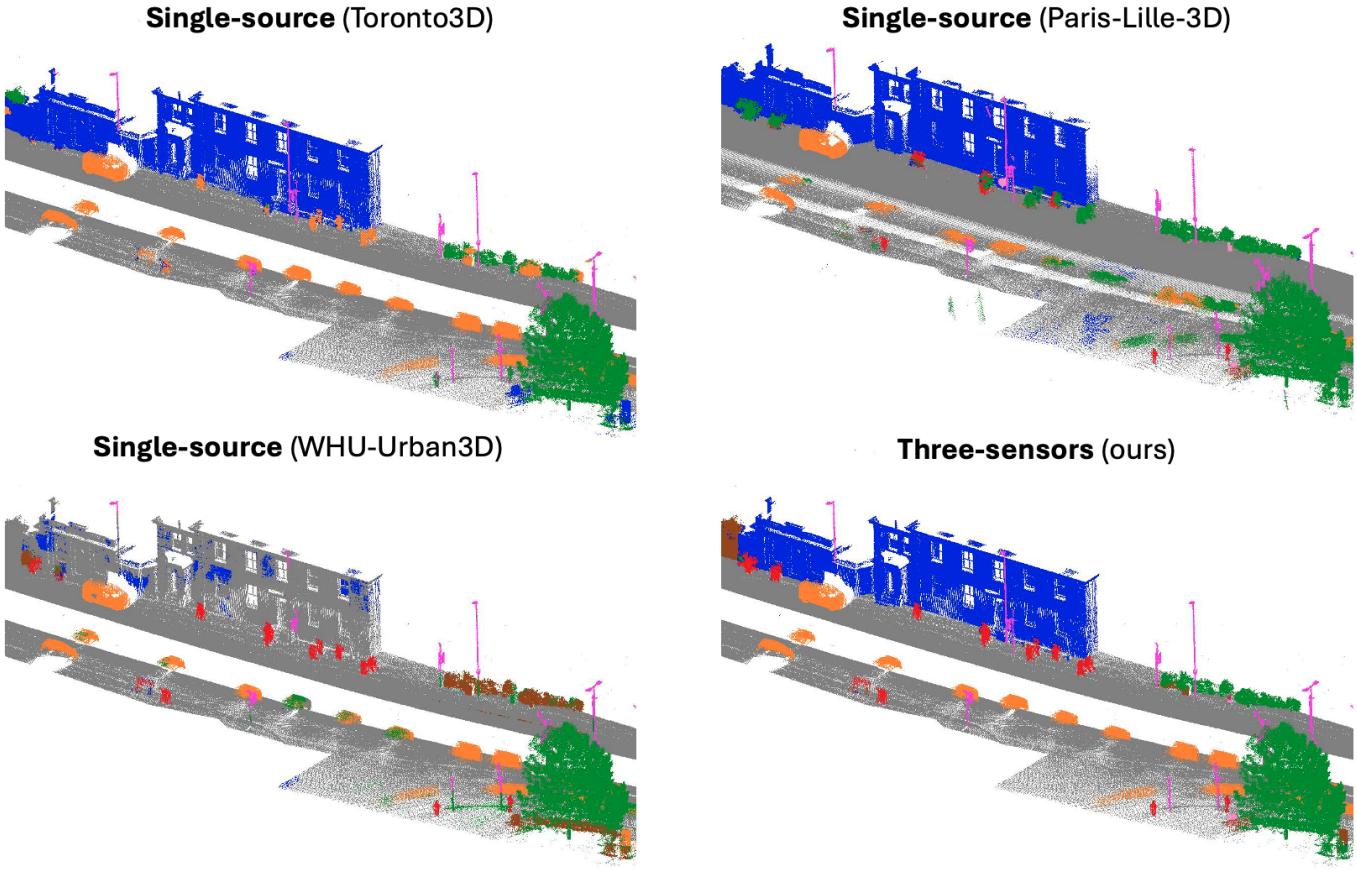

Figure 9. Deployment on the unlabelled OS point cloud. The scene is the one from Figure 1, now with the joint model added. Each single-source model fails as Figure 1 describes. The three-source model (Recipe B, coordinates only) recovers the scene, with buildings, road, cars, poles, and trees each in the right class (ground grey, building blue, natural green, pole magenta, car orange, pedestrian red).

For completeness we also ran a self-training step on the OS target. The effect on class shares was small. It changed only 2.4% of soton_l1 and 2.6% of soton_l2 points, mostly balanced churn along class boundaries, and left the class distribution essentially unchanged; on the labelled blocks it adds 0.010 mIoU (Table 14). This is expected. The target-free model is already confident on OS, keeping 92% of its own predictions above the 0.9 threshold, so self-training mostly reinforces them. Without ground truth, this alone cannot tell accurate predictions from confident errors; the labelled blocks below address this.

Ordnance Survey labelled three blocks of soton_l2 (up to 100 m × 100 m; 6.9 million raw points, 4.2 million 5 cm voxels). Labelling was done in ArcGIS Pro (version 3.6.0, Esri, Redlands, CA, USA): a coarse classification from rule-based tools and two pretrained Esri models was refined manually and checked against a 3D mesh from 7.5 cm aerial imagery and, for small objects, against the source frame imagery. The labels are therefore independent of our models but not fully manual. The blocks contain 1.40 million ground, 0.79 million building, 1.62 million natural, 38.5 thousand pole, and 0.10 million car voxels. We map ground, building, high vegetation, poles, and vehicles to the shared-5 classes and ignore all other codes, including noise. Every labelled voxel coincides with a point of the deployed inference chunks, so the models are scored exactly as deployed, with the same chunks and context. Besides shared-5 mIoU and per-class precision and recall, we report boundary mIoU, in the spirit of Boundary IoU [50], over the 0.24 million voxels that have a differently labelled point within 0.25 m, and full-resolution mIoU, where every raw point takes the prediction of its nearest voxel.

Table 14. Evaluation on the three labelled OS blocks (shared-5, 5 cm voxels in the deployed inference chunks, no TTA). Per-class IoU with precision/recall below. Boundary: voxels with a differently labelled point within 0.25 m. Full-Res.: raw points, nearest-voxel propagation. The self-trained model used the unlabelled OS cloud, including these blocks, for pseudo-labels.

Model	Ground	Building	Natural	Pole	Car	mIoU	Boundary mIoU	Full-Res. mIoU
Toronto3D only	0.965 0.99/0.97	0.881 0.89/0.99	0.934 0.99/0.94	0.805 0.97/0.82	0.812 0.85/0.94	0.879	0.478	0.889
Paris-Lille-3D only	0.849 0.99/0.85	0.746 0.75/0.99	0.944 0.98/0.96	0.814 0.98/0.83	0.596 0.99/0.60	0.790	0.545	0.814
WHU-Urban3D only	0.584 0.59/0.99	0.132 0.96/0.13	0.791 0.98/0.80	0.513 0.92/0.54	0.724 0.97/0.74	0.549	0.334	0.539
Avg single						0.739	0.452	0.747
Three-source DG (B)	0.980 0.99/0.99	0.944 0.96/0.98	0.964 1.00/0.97	**0.880** 0.98/0.90	0.956 0.99/0.97	0.945	0.665	0.954
DG + self-training on OS	**0.983** 0.99/1.00	**0.969** 0.98/0.99	**0.973** 1.00/0.98	0.867 0.99/0.88	**0.981** 1.00/0.98	**0.955**	**0.694**	**0.963**

Bold indicates the best IoU in each column.

Table 14 gives the results. The single-source models reach 0.879 (Toronto3D), 0.790 (Paris-Lille-3D), and 0.549 (WHU-Urban3D), an average of 0.739. The WHU-Urban3D model mislabels most building points as ground (building recall 0.13), the failure seen qualitatively in Figure 1. The three-source model reaches 0.945, above every single source, with pole at 0.880 and building at 0.944. Self-training on OS adds 0.010, to 0.955, so the target-free model reaches 99% of the self-trained score, consistent with the held-out sensors (Section 5.5). Boundaries remain the weak point. Boundary mIoU is 0.665 for the three-source model and 0.694 after self-training, with pole (0.43) and natural (0.54) lowest, mostly through low recall: thin objects lose their edge points to the larger class next to them. Full-resolution scores are within 0.01 of the voxel scores for every model except the Paris-Lille-3D-only model (0.790 voxel, 0.814 full resolution). With coord+int models standardized per cloud (Section 5.3), the single-source average on OS rises from 0.739 to 0.830, mostly through the WHU-Urban3D model (0.549 to 0.782). The multi-source scores on the blocks exceed the in-domain accuracy of the sources and should not be compared with Table 6; the blocks confirm the ordering of methods rather than the size of the cross-sensor gap. Three caveats apply. The three blocks come from one tile and one drive, so they are not a stratified sample of the city, and the semi-automatic labels may share some errors

with rule-based classification. The self-trained model also saw these blocks, unlabelled, during self-training.

6. Discussion

The cross-sensor gap is real and structured. Across the six transfer directions it averages about 0.24 shared-5 mIoU. The loss is uneven. Pole falls to 0.18 on Toronto3D and building to 0.33 on WHU-Urban3D, while ground and natural stay high. This is consistent with the reasoning in Section 3.3. A practical consequence is that in-domain accuracy is a weak guide to deployed accuracy.

Joint multi-sensor training is a simple, target-label-free remedy. It beats the average-single reference by 0.081 and, on the mean, exceeds the best-single oracle that a practitioner could not select. Its absolute gain is largest on Paris-Lille-3D (+0.122). On the hardest target, WHU-Urban3D, it exceeds even the best single source, by 0.052 in the reported run and 0.030 on the three-seed mean.

Target-free training captures most of the gain of a standard self-training step; stronger UDA methods may add more. A standard self-training step, which does use the unlabelled target, adds only 0.023 on top of joint training. Source combination therefore reaches about 97% of the final self-training score without touching the target, and it supplies 0.081 of the 0.104 total gain over the average-single reference. This makes it a strong, deployable reference. Dedicated cross-sensor UDA methods [13] should be measured by how much they add beyond it, not against a single-source starting point.

An ensemble of the single-source models performs on par with joint training (Section 5.6), so the benefit lies in combining sources with different failure modes rather than in a particular way of combining them. Confidence-based source selection, by contrast, picked the wrong source for WHU-Urban3D. A practitioner without target labels should therefore combine sources, by joint training or by ensembling.

The cross-sensor benefit of intensity required per-cloud standardization (Section 5.3), which assumes comparable intensity distributions across clouds. Range normalization, receiver gain, and scene content can violate this assumption. Radiometric calibration addresses this at the sensor level [51]. On the labelled OS blocks, per-cloud intensity raised the single-source average mostly through one single-run WHU-Urban3D model (Section 5.8). Its effect on the three-source model was not tested.

Two class and sensor pairs resist every coordinate-only method. Toronto3D pole stays between 0.15 and 0.18 and WHU-Urban3D building near 0.35 under single-source, joint, and self-training alike. Class weighting raises Toronto3D pole only to 0.27 and leaves WHU-Urban3D building unchanged (Section 5.7). Beam geometry is one plausible cause, but our design cannot fully separate the sensor from what travels with it (Table 3). Each source is also a different city, taxonomy, and annotation campaign, and the class balance differs sharply (Table 2). Building is 23% of labelled points in Toronto3D and 31% in Paris-Lille-3D but only 4.6% in WHU-Urban3D, whose scenes are dominated by vegetation, and pole is 1.6%, 0.55%, and 0.12% of the three sources. The WHU-Urban3D building failure is therefore consistent with a different sampling pattern and equally with a different urban form that offers few, differently shaped buildings. The Toronto3D pole failure may also reflect what the Toronto3D pole class contains, since the Paris-Lille-3D model, which reaches only 0.18 on Toronto3D pole, reaches 0.81 on OS pole (Table 14). The pair Toronto3D and Paris-Lille-3D shows that the acquisition setup matters beyond the scanner model. Their scanners are closely related (Table 3), yet transfer between them is strongly asymmetric (0.854 one way, 0.680 the other). The asymmetry must therefore come from other factors, such as the platform and mounting, which shape the scan pattern, or city, class balance, and annotation. Density alone is an unlikely explanation (Section 3.3). On the sensor side

this leaves the arrangement of samples, which depends on the mounting as much as on the scanner.

Separating sensor from city cleanly would need the same streets scanned by two sensors, which no public MLS dataset provides, and we regard this as the main open question behind the two hard cases.

Colour raises the same question as intensity. Toronto3D is the only source with colour and we dropped it to match the target, so we cannot test it here. We expect colour to behave more like intensity than like geometry. It depends on the camera, exposure, white balance, illumination, and season, all of which change between campaigns, so it is likely to need per-campaign normalization before it transfers, while remaining useful for reflectance-like classes such as road marking.

Our study has limits. The OS evaluation rests on three labelled blocks from one tile and one drive, with semi-automatic labels, so it confirms the main findings on the operational sensor but is not a stratified sample of the city. Labelled blocks across several drives and districts would make the deployment estimate more representative. Three labelled sensors form a controlled study rather than a large benchmark, and they let us compare one source with two but not more. Whether a fourth or fifth labelled sensor adds as much again is open, and the same protocol can answer it as datasets appear. Apart from the three-seed rows of Table 10 and the replicates in Table A2, the tables report one training run per model. Over three training seeds, joint training gains 0.079 ± 0.003 over the average-single reference and self-training adds a further 0.022 ± 0.003 (Table 10). Both gains exceed their seed-to-seed variation, so the ordering of methods is stable. Predictions are made at 5 cm voxel resolution and propagated to full resolution by nearest neighbour, so fine detail is approximate.

7. Conclusions

We presented a controlled study of cross-sensor generalization for MLS semantic segmentation across three labelled sensors and one unlabelled operational sensor. A 3×3 transfer matrix and a leave-one-sensor-out protocol quantified the gap. It is large and structured, concentrated on pole and building. Under a protocol suited to an unlabelled target, we compared against the average single source rather than an oracle. Joint multi-sensor training, using no target data, raised held-out shared-5 mIoU by 0.081, exceeded the best single source on the mean, and reached about 97% of the final score of a self-training step that does use the target. An ensemble of the single-source models performed on par with joint training, whereas confidence-based source selection was unreliable. Intensity helped reflectance classes in domain but transferred unevenly and only with target statistics, so geometry-only input was used for the main experiments. The result is a simple, deployable, target-free baseline for cross-sensor MLS segmentation, demonstrated end to end on a real Ordnance Survey point cloud, where it reached 0.945 shared-5 mIoU on three labelled blocks against 0.739 for the average single source. Because each source also differs in city, platform, and annotation, the gap we measure is that of the whole acquisition setting, not of the scanner alone. Future work should (i) design protocols that separate the scanner from city, platform, and annotation, for example the same scanner in several cities or several scanners on the same streets; (ii) audit taxonomy harmonization and annotation quality in multi-source benchmarks, and train robustly to the label noise that remains [52]; (iii) normalize and calibrate intensity across sensors [51]; (iv) select or ensemble models without target labels, supported by uncertainty estimates and confidence calibration [45–47]; (v) evaluate thin objects and boundaries at several resolutions [50], since voxelization and nearest-neighbour propagation blur them; (vi) build small labelled target sets that validate a deployment without labelling the whole area; and (vii) extend the study to more sensors, cities, and MLS configurations, for example with Paris-CARLA-3D [53].

Author Contributions: Conceptualization, J.M.G. and J.B.; methodology, J.M.G.; software, J.M.G.; validation, J.M.G.; formal analysis, J.M.G.; investigation, J.M.G.; resources, S.C., J.G. and J.B.; data curation, J.M.G., S.C. and J.G.; writing—original draft preparation, J.M.G.; writing—review and editing, Z.Z., S.C., J.G. and J.B.; visualization, J.M.G.; supervision, J.B., S.C. and J.G.; project administration, J.B.; funding acquisition, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Engineering and Physical Sciences Research Council (EPSRC) through an industrial CASE studentship with Ordnance Survey, grant numbers EP/X524840/1 and EP/W522077/1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The three labelled source datasets are publicly available: Toronto3D at https://github.com/WeikaiTan/Toronto-3D (accessed on 22 November 2026), Paris-Lille-3D at https://npm3d.fr/paris-lille-3d (accessed on 22 November 2026), and WHU-Urban3D at https://whu3d.com (accessed on 22 November 2026). The Ordnance Survey Southampton point cloud is a proprietary dataset collected for research purposes and is not publicly available. The preprocessing code, training configurations, and evaluation scripts are available at https://github.com/junegoo94/mls-cross-sensor-segmentation (accessed on 22 November 2026). The repository also links the trained weights of the single-source, leave-one-sensor-out, self-training, and three-source models.

Acknowledgments: The authors thank Ordnance Survey for providing the Southampton MLS point cloud and for their continued support of this research. During the preparation of this manuscript, the authors used Claude (Opus 5.5, Anthropic) for language editing and for the visual design of the methodology figure (Figure 3). The content and structure of Figure 3 were conceived and drafted by the authors; the AI tool was used only to refine its layout and visual styling. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: Stefano Cavazzi and James Gillespie are employees of Ordnance Survey, which co-funded this research through an EPSRC industrial CASE studentship and provided the unlabelled target dataset and the labels of the three evaluation blocks. They contributed to supervision, data provision, and manuscript review, and took part in the decision to publish the results, as described in the author contributions. Ordnance Survey had no other role in the design of the study; in the analyses or interpretation of data; or in the writing of the manuscript. EPSRC had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results. The authors declare no other conflicts of interest.

Appendix A. Class Mapping

Table A1 gives the mapping from the native classes of each source to the unified label space. Points mapped to ignore are excluded from training and evaluation. The less obvious decisions are the following. Toronto3D's road and the drivable and non-drivable ground of WHU-Urban3D become ground. Low vegetation and flower beds join trees in natural. Every pole-like class of WHU-Urban3D, including street lights and signal lights, becomes pole, following the Toronto3D convention in which the pole class includes lights and signs. WHU-Urban3D's box, board, detector, and other street furniture have no counterpart in the other sources and mix several object types, so they are ignored rather than forced into a unified class. WHU-Urban3D's box class also contains trash cans, but they cannot be separated from the other boxes and are therefore ignored too. Unclassified points are ignored in all sources.

Table A1. Native classes and their unified class. "ignore" means the points are excluded from training and evaluation. WHU-Urban3D classes are the 18 classes of the MLS-S partition (pywhu3d), each grouping several original codes.

Source	Native Class	Unified Class
Toronto3D	Road	Ground
	Road marking	Road marking
	Natural	Natural
	Building	Building
	Utility line	Utility line
	Pole	Pole
	Car	Car
	Fence	Fence/barrier
	Unclassified	ignore
Paris-Lille-3D	Ground	Ground
	Building	Building
	Pole, road sign, traffic light	Pole
	Bollard, small pole	Bollard
	Trash can	Trash can
	Barrier	Fence/barrier
	Pedestrian	Pedestrian
	Car	Car
	Natural, vegetation	Natural
	Unclassified	ignore
WHU-Urban3D	Non-driveway ground (incl. cycle lane, overpass, manhole)	Ground
	Driveway	Ground
	Tree	Natural
	Low vegetation (incl. flower bed)	Natural
	Building	Building
	Street light	Pole
	Telegraph pole	Pole
	Municipal pole	Pole
	Signal light	Pole
	Car	Car
	Road marking	Road marking
	Fence, road divider	Fence/barrier
	Wire	Utility line
	Person	Pedestrian
	Box (incl. trash can)	ignore
	Board (sign)	ignore
	Detector	ignore
	Other street furniture	ignore

Appendix B. Effect of the WHU-Urban3D v1.1 Correction

In the original release of WHU-Urban3D, a building about 22 m tall in validation scene 4629 (instance 27113116, 254,461 raw points) was annotated as pedestrian. Release v1.1 relabels it as building; comparing the two releases file by file, this is the only difference, and coordinates, intensity, and instance IDs are unchanged. No training scene is affected. Table A2 scores the same predictions on the WHU-Urban3D validation scenes with the original labels, where this instance is set to ignore, and with the corrected v1.1 labels, where it is building. Only building IoU changes, so every cross-sensor model gains about 0.015 shared-5 mIoU when WHU-Urban3D is the target, while the in-domain WHU-Urban3D score is unchanged (0.886). The gain of joint training over the average-single reference on this fold is 0.072 under both label sets. Models marked with an asterisk were retrained with an identical configuration and fixed seeds, because their checkpoints had not been kept for re-scoring with the v1.1 labels. Both columns come from one evaluation pass.

The original error was found incidentally while preparing the data. Afterwards we checked the pedestrian, pole, and car labels of all three sources for errors of the same kind.

We formed connected components of each class (0.3 m linking) and flagged pedestrians taller than 2.5 m, vehicles taller than 4 m, and any component wider than 15 m and taller than 2.5 m. Of about 9600 components (counted per overlapping 50 m chunk, so an object near a chunk border is counted more than once), only three distinct objects were flagged: a long truck in Toronto3D, and a bus and a truck in WHU-Urban3D. On visual inspection all three are correctly labelled vehicles, so we found no further errors of this kind. This check covers only large objects labelled as a small class; building labels themselves were not audited, and the check does not establish that the annotation is otherwise error-free.

Table A2. WHU-Urban3D validation, original vs. corrected (v1.1) labels: shared-5 mIoU, with building IoU in brackets. * Retrained with an identical configuration (see text).

Model	Original Labels	v1.1 Labels
Toronto3D only	0.638 [0.255]	0.653 [0.327]
Paris-Lille-3D only	0.598 [0.253]	0.613 [0.327]
WHU-Urban3D only (in domain)	0.886 [0.821]	0.886 [0.826]
DG, Recipe A unbalanced *	0.706 [0.304]	0.722 [0.384]
DG, Recipe A balanced *	0.683 [0.256]	0.698 [0.330]
DG, Recipe B	0.690 [0.271]	0.705 [0.347]
DG, Recipe B, seeds 2/3 *	0.646/0.667	0.662/0.683
DG + self-training	0.706 [0.267]	0.721 [0.342]
DG + self-training, seeds 2/3 *	0.684/0.679	0.700/0.694

Appendix C. Per-Class Results of the Intensity Experiment

Table A3 gives the per-class shared-5 IoU behind Table 7 for every cross-sensor transfer and input condition.

Table A3. Per-class shared-5 IoU of the cross-sensor intensity experiment (Section 5.3), same evaluation pass as Table 7. Gr = ground, Bu = building, Na = natural, Po = pole, Ca = car. Single training run per model.

Source → Target	Input	Gr	Bu	Na	Po	Ca	mIoU
Toronto3D → Paris-Lille-3D	Coord	0.949	0.949	0.803	0.628	0.958	0.857
	Per-cloud	0.925	0.953	0.807	0.739	0.978	0.880
	Source-only	0.904	0.920	0.783	0.734	0.926	0.853
Toronto3D → WHU-Urban3D	Coord	0.796	0.327	0.876	0.644	0.648	0.658
	Per-cloud	0.775	0.294	0.892	0.481	0.771	0.643
	Source-only	0.000	0.000	0.000	0.004	0.000	0.001
Paris-Lille-3D → Toronto3D	Coord	0.952	0.813	0.881	0.168	0.573	0.677
	Per-cloud	0.954	0.806	0.888	0.307	0.680	0.727
	Source-only	0.947	0.772	0.865	0.242	0.551	0.675
Paris-Lille-3D → WHU-Urban3D	Coord	0.798	0.327	0.912	0.437	0.581	0.611
	Per-cloud	0.781	0.307	0.884	0.276	0.615	0.573
	Source-only	0.000	0.000	0.000	0.002	0.001	0.001
WHU-Urban3D → Toronto3D	Coord	0.876	0.616	0.793	0.184	0.700	0.634
	Per-cloud	0.929	0.776	0.884	0.220	0.733	0.709
	Source-only	0.964	0.663	0.884	0.582	0.467	0.712
WHU-Urban3D → Paris-Lille-3D	Coord	0.634	0.315	0.733	0.395	0.812	0.578
	Per-cloud	0.876	0.869	0.682	0.393	0.897	0.743
	Source-only	0.943	0.596	0.413	0.386	0.416	0.551

References

1. Pan, J.; Cao, J.; Xing, S.; Dai, M.; Liu, J.; Wang, X.; Zhang, Y.; Huang, G. An aerial point cloud classification using point transformer via multi-feature fusion. *Sci. Rep.* **2025**, *15*, 22924. [CrossRef] [PubMed]
2. Wu, X.; Jiang, L.; Wang, P.S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; Zhao, H. Point transformer v3: Simpler, faster, stronger. In *Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2024; pp. 4840–4851.
3. Yan, H.; Lau, A.; Fan, H. Evaluating deep learning advances for point cloud semantic segmentation in urban environments. *KN-J. Cartogr. Geogr. Inf.* **2025**, *75*, 3–22. [CrossRef]
4. Xiao, W.; Vallet, B.; Paparoditis, N. Change detection in 3D point clouds acquired by a mobile mapping system. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *2*, 331–336. [CrossRef]
5. Aijazi, A.; Checchin, P.; Trassoudaine, L. Detecting and updating changes in lidar point clouds for automatic 3d urban cartography. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *2*, 7–12. [CrossRef]
6. Kissling, W.D.; Shi, Y.; Koma, Z.; Meijer, C.; Ku, O.; Nattino, F.; Seijmonsbergen, A.C.; Grootes, M.W. Country-wide data of ecosystem structure from the third Dutch airborne laser scanning survey. *Data Brief.* **2023**, *46*, 108798. [CrossRef] [PubMed]
7. Yi, L.; Gong, B.; Funkhouser, T. Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2021; pp. 15358–15368.
8. Kim, J.; Woo, J.; Kim, J.; Im, S. Rethinking lidar domain generalization: Single source as multiple density domains. In *Proceedings of the European Conference on Computer Vision*; Springer: Cham, Switzerland, 2024; pp. 310–327.
9. Goo, J.M.; Zeng, Z.; Boehm, J. Zero-Shot Detection of Buildings in Mobile LiDAR using Language Vision Model. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *XLVIII-2-2024*, 107–113. [CrossRef]
10. Goo, J.M.; Zeng, Z.; Boehm, J. Flash: Real-Time Lidar Super-Resolution Via Frequency-Aware Multi-Scale Fusion. In *Proceedings of the 2026 IEEE International Conference on Image Processing (ICIP)*; IEEE: Piscataway, NJ, USA, 2026; pp. 1–6.
11. Goo, J.M.; Zeng, Z.; Boehm, J. Sensor Domain Adaptation for 3D Object Detection via LiDAR Super-Resolution. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2026**, *XI-2-2026*, 93–100. [CrossRef]
12. Goo, J.M.; Zeng, Z.; Boehm, J. A Comprehensive Survey on Deep Learning-Based LiDAR Super-Resolution for Autonomous Driving. In *Proceedings of the 2026 IEEE Intelligent Vehicles Symposium (IV)*; IEEE: Piscataway, NJ, USA, 2026; pp. 2016–2023. [CrossRef]
13. Luo, H.; Chen, Z.; Ye, F.; Huang, T.; He, H.; Hu, W. Cross-sensor adaptive semantic segmentation for mobile laser scanning point clouds based on continuous potential scene surface reconstruction. *ISPRS J. Photogramm. Remote Sens.* **2025**, *228*, 537–551. [CrossRef]
14. Sanchez, J.; Deschaud, J.E.; Goulette, F. COLA: COarse-LAbel Multisource LiDAR Semantic Segmentation for Autonomous Driving. *IEEE Trans. Robot.* **2025**, *41*, 1742–1754. [CrossRef]
15. Sanchez, J.; Deschaud, J.E.; Goulette, F. 3DLabelProp: Geometric-Driven Domain Generalization for LiDAR Semantic Segmentation in Autonomous Driving. *arXiv* **2025**, arXiv:2501.14605.
16. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2017; pp. 652–660.
17. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5099–5108.
18. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2019; pp. 6411–6420.
19. Choy, C.; Gwak, J.; Savarese, S. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 3070–3079.
20. Graham, B.; Engelcke, M.; Van Der Maaten, L. 3D semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 9224–9232.
21. Milioto, A.; Vizzo, I.; Behley, J.; Stachniss, C. Rangenet++: Fast and accurate lidar semantic segmentation. In *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Piscataway, NJ, USA, 2019; pp. 4213–4220.
22. Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Ma, Y.; Li, W.; Li, H.; Lin, D. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2021; pp. 9939–9948.
23. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2021; pp. 16259–16268.

24. Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; Zhao, H. Point transformer v2: Grouped vector attention and partition-based pooling. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 33330–33342. [CrossRef]
25. Pointcept Contributors. Pointcept: A Codebase for Point Cloud Perception Research. 2023. Available online: https://github.com/Pointcept/Pointcept (accessed on 22 November 2026).
26. Tan, W.; Qin, N.; Ma, L.; Li, Y.; Du, J.; Cai, G.; Yang, K.; Li, J. Toronto-3D: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*; IEEE: Piscataway, NJ, USA, 2020; pp. 202–203.
27. Han, X.; Liu, C.; Zhou, Y.; Tan, K.; Dong, Z.; Yang, B. WHU-Urban3D: An urban scene LiDAR point cloud dataset for semantic instance segmentation. *ISPRS J. Photogramm. Remote Sens.* **2024**, *209*, 500–513. [CrossRef]
28. Roynard, X.; Deschaud, J.E.; Goulette, F. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int. J. Robot. Res.* **2018**, *37*, 545–557. [CrossRef]
29. Zeng, Z.; Boehm, J. Exploration of an open vocabulary model on semantic segmentation for street scene imagery. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 153. [CrossRef]
30. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. Semantic3D. net: A new large-scale point cloud classification benchmark. *arXiv* **2017**, arXiv:1704.03847.
31. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2019; pp. 9297–9307.
32. Fong, W.K.; Mohan, R.; Hurtado, J.V.; Zhou, L.; Caesar, H.; Beijbom, O.; Valada, A. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robot. Autom. Lett.* **2022**, *7*, 3795–3802. [CrossRef]
33. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. Nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2020; pp. 11618–11628.
34. Hu, Q.; Yang, B.; Khalid, S.; Xiao, W.; Trigoni, N.; Markham, A. Towards semantic segmentation of urban-scale 3D point clouds: A dataset, benchmarks and challenges. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2021; pp. 4975–4985.
35. Rist, C.B.; Enzweiler, M.; Gavrila, D.M. Cross-sensor deep domain adaptation for lidar detection and segmentation. In *Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV)*; IEEE: Piscataway, NJ, USA, 2019; pp. 1535–1542.
36. Li, M.; Zhang, Y.; Ma, X.; Qu, Y.; Fu, Y. BEV-DG: Cross-Modal Learning under Bird's-Eye View for Domain Generalization of 3D Semantic Segmentation. In *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Los Alamitos, CA, USA, 2023; pp. 11598–11608. [CrossRef]
37. Prieur, J.F.; St-Onge, B.; Fournier, R.A.; Woods, M.E.; Rana, P.; Kneeshaw, D. A Comparison of Three Airborne Laser Scanner Types for Species Identification of Individual Trees. *Sensors* **2022**, *22*, 35. [CrossRef] [PubMed]
38. Kong, L.; Quader, N.; Liong, V.E. ConDA: Unsupervised Domain Adaptation for LiDAR Segmentation via Regularized Domain Concatenation. In *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)*; IEEE: Piscataway, NJ, USA, 2023; pp. 9338–9345. [CrossRef]
39. Saltori, C.; Galasso, F.; Fiameni, G.; Sebe, N.; Poiesi, F.; Ricci, E. Compositional semantic mix for domain adaptation in point cloud segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14234–14247. [CrossRef] [PubMed]
40. Kim, H.; Kang, Y.; Oh, C.; Yoon, K.J. Single domain generalization for lidar semantic segmentation. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2023; pp. 17587–17598.
41. Griffiths, D.; Boehm, J. Weighted Point Cloud Augmentation for Neural Network Training Data Class-Imbalance. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W13*, 981–987. [CrossRef]
42. Bayrak, O.C.; Ma, Z.; Farella, E.M.; Remondino, F.; Uzar, M. Combining 3D Urban Objects from All Around the World to Improve Object Classification and Semantic Segmentation. *PFG—J. Photogramm. Remote Sens. Geoinf. Sci.* **2026**, *94*, 305–332. [CrossRef]
43. Berman, M.; Triki, A.R.; Blaschko, M.B. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2018; pp. 4413–4421.
44. Nekrasov, A.; Schult, J.; Litany, O.; Leibe, B.; Engelmann, F. Mix3D: Out-of-context data augmentation for 3d scenes. In *Proceedings of the 2021 International Conference on 3D Vision (3DV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 116–125.
45. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6402–6413.
46. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In *Proceedings of the International Conference on Machine Learning*; PMLR: New York, NY, USA, 2017; pp. 1321–1330.
47. Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.; Lakshminarayanan, B.; Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 13991–14002.

48. Cui, Y.; Jia, M.; Lin, T.Y.; Song, Y.; Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2019; pp. 9260–9269.
49. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*; IEEE: Piscataway, NJ, USA, 2017; pp. 2980–2988.
50. Cheng, B.; Girshick, R.; Dollár, P.; Berg, A.C.; Kirillov, A. Boundary IoU: Improving object-centric image segmentation evaluation. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2021; pp. 15329–15337.
51. Kashani, A.G.; Olsen, M.J.; Parrish, C.E.; Wilson, N. A review of LiDAR radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration. *Sensors* **2015**, *15*, 28099–28128. [CrossRef] [PubMed]
52. Ye, S.; Chen, D.; Han, S.; Liao, J. Learning with noisy labels for robust point cloud segmentation. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: Piscataway, NJ, USA, 2021; pp. 6423–6432.
53. Deschaud, J.E.; Duque, D.; Richa, J.P.; Velasco-Forero, S.; Marcotegui, B.; Goulette, F. Paris-CARLA-3D: A real and synthetic outdoor point cloud dataset for challenging tasks in 3D mapping. *Remote Sens.* **2021**, *13*, 4713. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.