Improving Rare Classes on nuScenes LiDAR segmentation Through Targeted Domain Adaptation

Vickram Rajendran, Chuck Tang, Frits van Paasschen
Applied Intuition
Mountain View, CA, 94041
{vickram, chuck.tang, frits}@applied.co

Abstract

We generate synthetic data in order to target improvement on specific rare classes in LiDAR segmentation without regressing performance on the existing, plentiful classes. While using auxiliary data to improve performance on a domain is not new with respect to classification, there is limited research on targeting specific classes with this technique. It is currently unclear how to extend those methods to work well on more complicated but realistic use cases for autonomous driving such as LiDAR segmentation.

By upsampling specific classes in the auxiliary domain, mixing data between domains, and splitting representation building and fine-tuning, we are able to see impressive improvements on a targeted rare class without losing performance on the other classes. On the popular autonomous driving benchmark nuScenes, we use this procedure to improve performance on the rare class of cyclists by 18%, resulting in the best Cylinder3D model on the LiDAR segmentation benchmark. We also show that these techniques extend to other classes (debris) and other tasks (LiDAR object detection), giving strong evidence that this methodology generalizes well to other autonomous perception tasks.

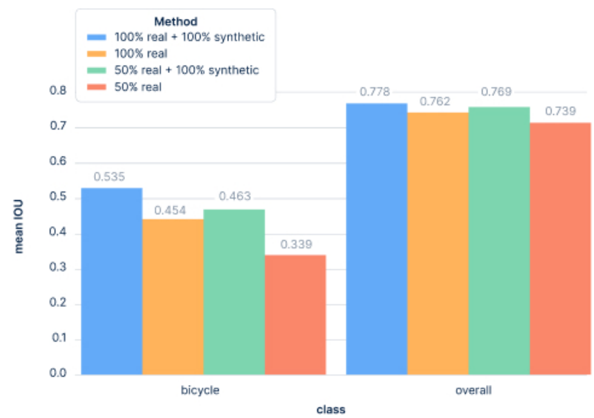


Figure 1. Using the Log Mod 60 Synthetic nuScenes dataset to improve performance on bicycles for nuScenes LiDAR segmentation with the data and training strategy proposed in this study.

1. Introduction

Training highly performant perception models for a particular domain is a necessity for many real-world applications of machine learning, such as autonomous vehicles. While these models have improved significantly in the past several years of research, unfortunately, they still require a significant amount of data for each class and can struggle to perform well on rare classes. In particular, most real-world datasets are significantly long-tailed, which presents challenges for perception model training for two reasons. First, perception models require large amounts of data to generalize well. Second, standard perception models will tend to focus on classes that have more representation.

There are many techniques that have been developed to deal with long-tailed datasets, whether it is by changing the model's loss function, re-balancing the data, or modifying the architecture [24] [19] [8]. In this work, we focus on using auxiliary data of the same class from a different domain in order to improve performance on the rare class (e.g., collecting more data or using data from a different dataset that has examples of that class). Unfortunately, the auxiliary data often has an implicit **domain gap** between the target domain and the auxiliary domain (e.g., if we used data from KITTI [14] to improve performance on a rare nuScenes [6] class). These domain gaps may even be entirely unknown – perhaps the sensors, environments, or actors have imperceptibly changed during the collection of a second dataset, to the point where we may not even know that there was a domain gap at all. This domain gap can make the auxiliary data far less useful, and in some cases, deteriorate the original model performance.

As seen in [2], naively training on a combined dataset of the two domains does not improve performance on rare classes. More advanced techniques are needed in order to make use of auxiliary datasets. While there has been some work on this topic in the classification setting [3] [11], it is unclear how one would modify those techniques to work on real-world perception tasks for autonomous driving like LiDAR semantic segmentation, which presents some task-specific challenges. For example, semantic segmentation data often consists of multiple classes (and often not just the rare class), as well as potentially larger domain gaps compared to classification due to increased reliance of the models on context and background [23]. To the best of our knowledge, we are the first to explore this setting for tasks other than image classification.

Our contributions are fourfold:

- We demonstrate a method to target rare classes by generating an auxiliary synthetic dataset based on the original labeled data, and then modifying that synthetic dataset in order to improve performance on specific classes of interest.
- We develop a training procedure that uses auxiliary data to improve rare classes for LiDAR segmentation. Further, we show how to extend recent work in group-invariant learning for classification to domain adaptation in LiDAR segmentation, and show that these techniques generalize to different classes and tasks.
- We improve the performance of the best Cylinder3D model significantly on the nuScenes-lidarseg benchmark with these techniques, improving the mean intersection over union score (mIOU) by 13 points on the bicycle class.
- We show that training with this auxiliary data as well as only 50% of the nuScenes dataset achieves the same performance as the baseline model, showing that the auxiliary data can act as a reasonable replacement for a large proportion of real data.

2. Related Work

Improving rare classes: Most work in domain adaptation is focused on improving all classes in the target domain [30]. There is significantly less work oriented towards improving specific classes with data from a different domain. [3] does this for classification by adding synthetically generated examples of rare classes into the training set and using domain adaptive methods such as DANN [20] and CORAL [28] in order to improve performance on a subclass of deer; however, their discussion is focused on classification. [8] utilizes a method similar method to ours, which generates and augments the existing dataset with hybrid points; however, their method does not extend well to higher-dimensional problems in computer vision. Further, in our method, we generate an entirely new dataset and mix data from our new domain and the target domain. [7] pursues a similar method called Partial Domain Adaptation. They try to take a large-scale source domain and improve performance on a smaller-scale target domain. Our setting is a slight generalization from this, as we assume that we have labeled data from the target domain for all classes, even if it is only rare classes for some of them. To the best of our knowledge, there are no examples of using auxiliary data to target a specific rare class for LiDAR segmentation.

Domain Adaptation and Generalization: There has been significant amounts of work in this field. Our setting is closest to unsupervised domain adaptation [31] [25] [34], where there are no labels in the target domain. In contrast, we assume we have lots of data in the target domain for some classes but very little for specific rare classes. This is particularly important as recent work has found that many unsupervised domain adaptation methods are completely outperformed by training with ordinary supervised learning methods with fewer than 70 images from the target domain [26]. Improving performance when there is little real data in the target domain is thus highly valuable. It is unclear whether data from one domain can be used to improve performance in this few-shot single-class case. [13] attempts to use targeted augmentations that rely on prior knowledge to determine exactly which variations the model should be robust to; however, it is unclear how one would generalize these techniques to non-classification datasets.

Group Invariant Learning: There has been a recent focus on training models that have strong performance across all subgroups of the data, e.g. ensuring that performance on aerial waterbirds is the same as land waterbirds [27], even when there is a spurious correlation in the training dataset. Our work is similar in that we have two subgroups (domains), but we are interested in the case where we have very little data in one subgroup and are using the other subgroup to improve performance on the first subgroup. Our work can be considered as an attempt at adding a different group to the training set along with performing group invariant learning in order to improve performance on the original domain. To the best of our knowledge, we are the first to extend these group-invariant techniques to 3D perception tasks for autonomous driving.

3. Methodology

We walk through both the data generation as well as the training process that we use in order to target performance on rare classes. For the remainder of this study, we will focus on the nuScenes LiDAR segmentation task and target the rare class of bicycles.

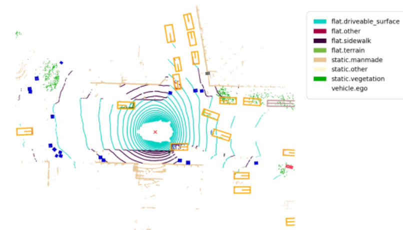
nuScenes is a large-scale autonomous driving dataset by Motional collected from real-world drive logs in Boston and Singapore with camera, LiDAR, and radar sensors. The dataset contains multiple public benchmarks such as LiDAR segmentation [12] and LiDAR object detection, with full annotations. NuScenes contains 5.5 hours of data, including over 78 million labeled LiDAR points and over a million cuboids of real-world objects. Annotations range from dynamic actors such as cars and motorcycles to static obstacles such as vegetation and driveable surfaces. Since nuScenes was collected in the real world, it suffers from class imbalances in its data distribution, which makes it a prime **target dataset** for our work. In this study, we will focus on the **bicycle** class as an example of a rare class that we will target. As seen in Figure 3, bicycles have an imbalance of over 1,000:1 with respect to the other classes in LiDAR segmentation points.

We show our main methodology for improving the bicycle class in Figure 4 and dive into each component in the following sections.

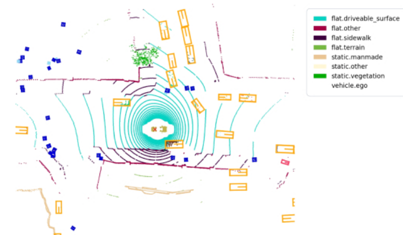
3.1. Generating Auxiliary Datasets

Determining the right auxiliary dataset is challenging. In order to train on two datasets jointly, the datasets need to be similar enough to avoid negative transfer from training on them together. Unfortunately, this can be difficult since few datasets share the exact same class taxonomy, labeling procedures, and annotations. For example, the SemanticKITTI [4] dataset has significantly different annotations and annotation guidance than the nuScenes dataset. In order to use information from SemanticKITTI to improve performance on nuScenes, one needs to either treat these datasets as entirely separate and use different loss functions or use one of them as a source domain for pre-training before training on the target domain. To address this problem, we generate our own auxiliary datasets to improve performance on nuScenes.

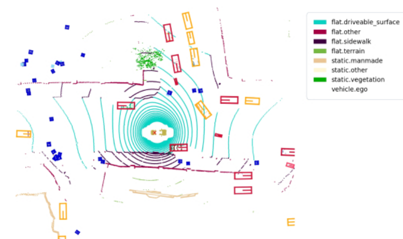
We use the Applied Intuition Synthetic Datasets software [1] to generate and modify a synthetic recreation of the nuScenes training dataset. First, we generate a synthetic copy of the nuScenes LiDAR segmentation training dataset, matching actor placements and behaviors with the real nuScenes dataset. An example of this is shown in Figure 2. This allows us to ensure that the same annotation spec and behaviors are in the auxiliary dataset, removing any potential domain gaps that could occur due to actor behavior and placement. The key advantage of building a dataset in simulation is the ability to modify it to target specific classes that we wish to improve on. In order to focus on improvement for bicycles specifically, we upsample bicycles in the synthetic recreation by replacing a proportion of car assets with bicycle assets. We expect that replacing cars with bikes will have a smaller domain gap than randomly placing bikes in the scene placement is more likely to be realistic. Due to the large number of cars in the real dataset, we do not expect this replacement to harm performance.



(a) A real nuScenes LiDAR scan



(b) A synthetic recreation of the LiDAR scan above



(c) A modified synthetic recreation of the first LiDAR scan with cyclists replacing cars

Figure 2. Example LiDAR scans and annotations from the generated synthetic data for nuScenes scene 103.

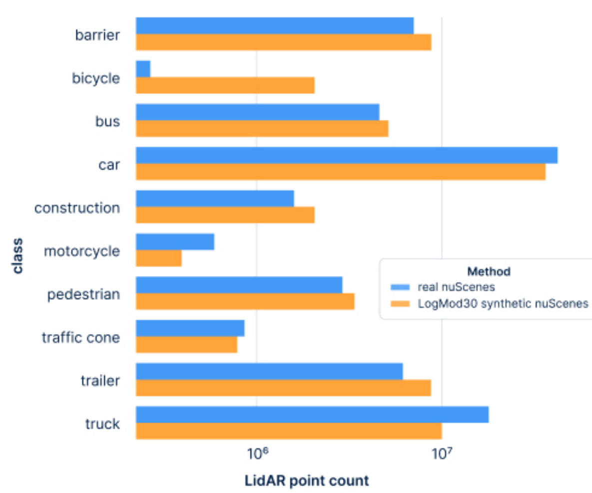


Figure 3. LiDAR segmentation class counts (log-scale) for the original nuScenes dataset and the Log Mod 30 Synthetic nuScenes dataset. Note the trade-off between cars and bicycles in the synthetic dataset.

During the upsampling process, we use six different bicycle assets to replace examples of cars and perform domain randomization [29] [9] on the asset dimensions and LiDAR sensor noise floor to increase the diversity of the auxiliary synthetic nuScenes dataset. To determine the effect of this replacement, we generate three datasets with varying percentages of cars replaced (30%, 60%, and 90%). Figure 3 shows the class distributions of the dataset with 30% replacement. We call these datasets **Log Mod Datasets**, as we are taking the real drive logs and modifying them to have more bicycles. We use these datasets as well as the unmodified synthetic nuScenes as the auxiliary datasets in the remainder of this study.

3.2. Training Strategy

Once we have our auxiliary dataset, we must determine the right way to train with it. The naive thing to do is to train on a combination of the two domains. We can either combine all of the data together or join the portions from the auxiliary domain that contain the underrepresented classes. This setting has been studied for classification in [2], showing that when the target domain contains no examples of the underrepresented class, joint training (even with domain adaptation) fails significantly worse than when using the auxiliary domain and ignoring the target domain entirely.

In Figure 5, we confirm that these results continue to hold even in the segmentation case and that adding auxiliary data does not significantly improve the performance on rare classes in the target domain. While we do see some improvement on the bicycle class, the domain gap is large enough that the model struggles to make significant gains on the class of interest. To mitigate this issue, we make several changes to the loss function and training strategy in order to make better use of the auxiliary data.

Inspired by recent work in image classification that separates the representation and classification stage of a model to deal with class imbalance and subgroup performance [16] [17], we consider this problem in two stages. First, we train with the auxiliary domain in order to build strong representations of the bicycle class. Second, we fine-tune this pre-trained model on the target domain only to remove any spurious features from the auxiliary domain. We follow recent research showing that fine-tuning can distort pre-trained features [18] and follow the process of first training the head only (linear probe), and then fine-tuning the whole representation in order to focus the final layers on the target domain's task. This procedure is very similar to other methods such as [16] and [17], which freeze the representation in order to perform well on imbalanced data, and we hypothesize that combining these results will allow the models to get the best use out of the auxiliary domain.

Extending this method from classification to segmentation is non-trivial due to a number of factors:

- Will this method work on a semantic segmentation task where features consist of multiple classes across the whole image, as opposed to classification where features are associated with a single class?
- How well do these techniques address domain gaps in the initial pre-training, since none of the existing research does the representation-building with multiple domains?
- How do we deal with the fact that we cannot provide a balanced reweighting dataset as in [17], due to the fact that segmentation datasets consist of already imbalanced data?

We explore these issues in the following sections.

3.3. Building Strong Representations

We employ a number of strategies to improve the representation capability of the models with the auxiliary data. The natural approach is to train on the two datasets together and use the representation built with this. However, we have seen in Figure 5 that this does not perform well.

To mitigate this, we use **mixing strategies**, specifically PolarMix [32] for the LiDAR segmentation task. These have been empirically shown to improve performance on unsupervised domain adaptation for LiDAR segmentation as well as 2D object detection and semantic segmentation [35] [22]. We hypothesize that these methods work by building stronger and less distinguishable domain representations during the training process, which allows the models to see the auxiliary rare classes in many different target domain contexts. [15] uses a similar technique in order to bridge the domain gap between different LiDAR sensors, which further bolsters that conclusion. Since we assume that we have at least some data in the target domain of the rare class, we do not expect to run into the same problems as [2], where using Mixup between domains that shared no classes did not improve performance. Following [16] and [17], we do not make any modifications to the loss functions or the training process during the joint training of the two datasets, as they find that traditional empirical risk minimization is sufficient to build good representations.

3.4. Fine-tuning the Representation

Following [18], we perform a strategy of linear probing and then fully fine-tuning (LP-FT) in order to fine-tune the representation. First, we freeze the representation and train just the head of the model. For segmentation, this is a simple task as the representation is all but the last layer. We fine-tune the model on the **target dataset only** and use the auxiliary dataset only for the joint representation-building. This ensures that the final model does not hold on to irrelevant information from the auxiliary domain and helps address any potential negative transfer from the domain gap.

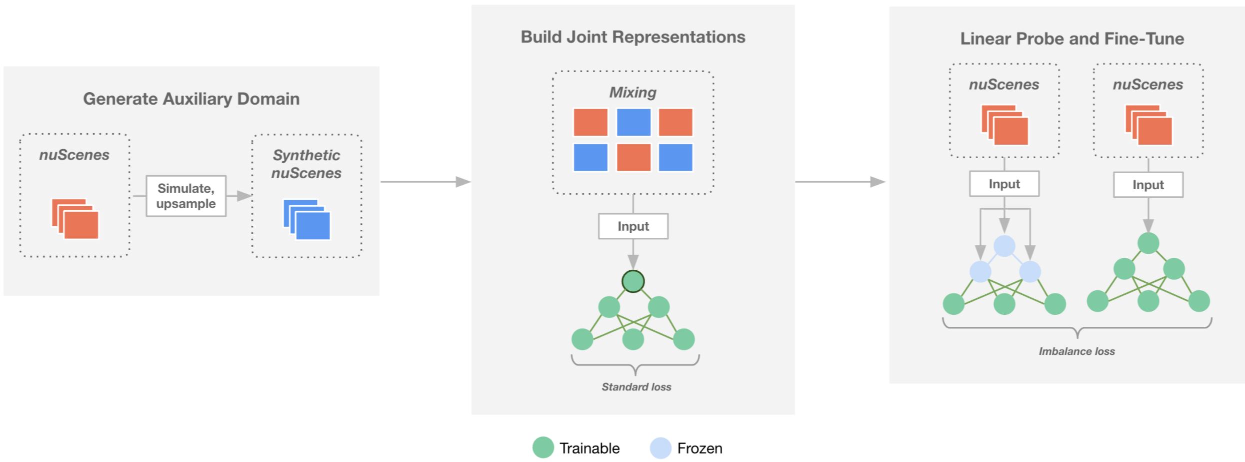


Figure 4. Our proposed methodology to target rare classes for nuScenes LiDAR segmentation.

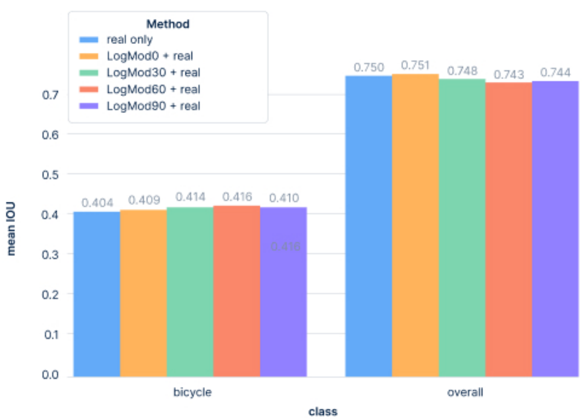


Figure 5. Rare class performance on LiDAR segmentation after naive joint training with the auxiliary datasets. Note that joint training does not significantly improve performance on the rare class, even though bicycles are upsampled.

Since we have a rare class, we would like to use a strategy like Deep Feature Reweighting (DFR) [17] in order to make the most of the strong representation. However, this requires a held-out balanced reweighting dataset which we cannot receive in the segmentation case due to its inherent imbalance. Instead, we train with a class-weighted loss function (where the class counts are based on the target dataset) in order to focus training on the rare class during the fine-tuning process. While there are other methods of dealing with the imbalance, we find a straightforward modification to the loss function to be a first easy choice to use. We find that using this modified imbalance strategy still improves performance and is very easy to adapt to the object detection and segmentation use case, compared to DFR, which uses a balanced reweighting dataset. We use this class-imbalanced loss for both the head retraining as well as the full fine-tuning.

Figure 4 summarizes our training strategy. We do a three-step approach. First, we generate an auxiliary dataset of our original target domain dataset and modify it to upsample the class we wish to target. Then, we pre-train the model on both the auxiliary and the original dataset with a mixing strategy designed to encourage shared strong representations. Finally, we perform linear probing and train the head of the model, before fine-tuning our model, with both of these final training steps happening on the target data and using a class-weighted imbalanced loss.

4. Experimental Setup

4.1. Training Details

For the LiDAR segmentation experiments, we use the popular Cylinder3D architecture and codebase [36]. We follow their settings and train models for 40 epochs at a learning rate of 0.001 with cosine decay. Note that this means that models which jointly train on the synthetic and real data will see the same amount of real data as models which train on real data only. We use AdamW [21] as the optimizer for all experiments. We use a sum of the Lovász-Softmax [5] as well as cross-entropy (in some cases weighted during the imbalance experiments) in order to train the model. For all experiments, we have a batch size of 6 and use two Nvidia A5000 GPU's.

For object detection, we use the Centerpoint [33] architecture through mmdetection3D [10]. Our model jointly trains on 100 epochs for pre-training and then fine-tunes for a further 50 epochs. The learning rate is cyclic between $5e^{-5}$ and $5e^{-4}$ every 20 epochs. We have a batch size of 16 and train with eight Nvidia Tesla v100s.

We use the official nuScenes training and validation splits for both the segmentation and the detection tasks. For all cases, we generate the auxiliary domain by creating a synthetic copy of **only** the nuScenes training splits. Our validation dataset is always the official nuScenes validation split with no additions or modifications. Models train from scratch, without any pre-trained weights from other datasets.

4.2. Evaluation

We are purely interested in the performance improvement on rare classes due to the auxiliary domain, but the techniques we are using (such as fine-tuning, linear probing, etc.) may also improve the performance of the target-only baseline model. Thus, in order to perform a fair evaluation, we perform the exact same experiments and training strategies with a model that has only seen the target domain data and does not have access to any auxiliary data.

We report performance on the specific class we are interested in (i.e., bicycles), as well as an aggregate metric over all of the classes to ensure that improvements in the targeted class do not cause regressions on the other classes. We use the nuScenes validation dataset for all of our results. For semantic segmentation we use the mIOU metric. For object detection, we use the nuScenes detection score (NDS) [6].

5. Experiments

5.1. Representation Building

First, we conduct a series of experiments that test the efficacy of the mixing strategy and whether or not using the auxiliary domain actually helps in building a good representation. We train two different LiDAR segmentation models for each auxiliary dataset, and three models with no auxiliary dataset. The first model trains normally on all of the data, with standard data augmentations such as flipping, translation, and rotation of the LiDAR point clouds. The second model adds PolarMix data augmentation during the joint training process. All other experimental details such as architecture, epochs, etc. remain the same.

Figure 6 shows the result of the joint training procedure with and without PolarMix for the best dataset. While PolarMix improves both the real-only model and the models with auxiliary data, it disproportionately improves the performance when using data from a different domain. This is evidence that the existence of a different domain, especially where we have upsampled the number of a rare class, improves the representation capabilities of that class significantly.

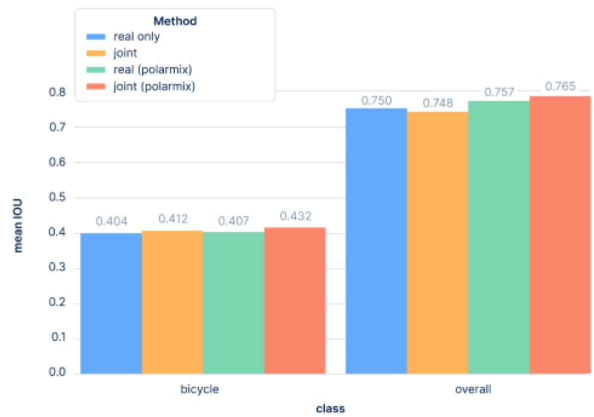


Figure 6. Performance of joint training with the Log Mod 30 Synthetic nuScenes dataset with and without PolarMix data augmentation in the representation building. Note that PolarMix improves the joint models much more than it improves the real models.

5.2. Fine-tuning Performance

Once we have built good representations, we perform the LP-FT strategy with the nuScenes dataset to focus the models on the target domain. For the linear probing in Cylinder3D, we freeze all layers except for the final layer of the last MLP. For both the linear probing and the full fine-tuning, we use weights based off of the square root of the class counts in order to determine the class weights for the weighted cross-entropy. We do not change the Lovász-Softmax portion of the loss. We keep the learning rate schedule and the weight decay constant from pre-training and train each stage of the LP-FT process for 40 epochs before picking the model that has the lowest validation loss.

The result of these experiments is shown in Table 1. First, we find that up to a certain point, increasing the proportion of the rare, upsampled class improves performance on the rare class significantly. However, when we reach 90% of cars replaced with bikes in the auxiliary dataset, we find that improvement sharply decreases. This is likely because the model has so little signal from the auxiliary dataset that we are simply experiencing negative transfer, and the performance degrades to not having any auxiliary data at all.

Note that this technique also increases the performance of the nuScenes-only trained model, providing a mIoU benefit of 1.2 points and improving the bike score by 5 points. However, the best-performing model with auxiliary data further increases the bicycle score **by 8 points of mIoU**. It even improves performance on the other classes as the overall mIoU improves by 1.6 points. We attribute this to the general addition of synthetic examples of those classes, even if we did not target those classes specifically. To the best of our knowledge, this is now the best-performing

Table 1. LiDAR segmentation performance with our methodology on the nuScenes validation dataset.

Model	Barrier	Bicycle	Bus	Car	Construction	Motorcycle	Pedestrian	Traffic Cone	Trailer	Truck	Drivable	Other	Sidewalk	Terrain	Manmade	Vegetation	Average
Real Baseline	76.5	40.4	92.3	87.0	46.3	79.2	77.0	65.6	62.1	81.0	96.3	74.3	73.8	74.0	88.1	86.6	75.0
Real PolarMix LP-FT	76.6	45.4	93.5	87.3	52.3	82.7	77.2	68.2	66.8	76.7	96.5	70.7	74.2	74.2	89.4	87.3	76.2
Joint PolarMix LP-FT 0	76.7	48.9	93.1	90.2	53.1	87.0	80.7	65.3	66.6	85.0	96.5	72.4	75.2	75.2	89.4	87.8	76.7
Joint PolarMix LP-FT 30	76.9	52.2	93.4	90.4	54.9	86.0	81.5	67.9	67.2	85.0	96.6	71.4	74.4	74.6	89.5	87.9	78.1
Joint PolarMix LP-FT 60	76.5	**53.5**	93.3	88.0	52.4	85.4	82.1	67.9	64.9	84.2	96.7	71.3	75.5	75.6	89.2	87.8	77.8
Joint PolarMix LP-FT 90	77.0	51.4	94.6	87.7	54.5	86.4	78.4	65.1	70.7	84.6	96.7	72.0	75.9	75.5	89.2	88.0	77.2

Cylinder3D model on the nuScenes dataset, with performance matching newer and much more complicated architectures and training strategies.

To study the effect of each individual component of the training strategy, we show the results of an ablation study in Table 2. From this, we can see that, while performance on the real data improves with these techniques (LP-FT, square-root sampled weighted cross entropy, PolarMix), the benefits of the auxiliary dataset are disproportionately improved by these same techniques. While the model that trained only on the target domain improves the performance on the bicycle class by 5 points, using the auxiliary domain further improves the model by another 8 points on bicycles. Secondly, we can see that all three of these techniques provide modest improvements to model performance, and can be used independently of each other. This is particularly useful information, as some techniques may be unable to be used depending on the exact use case for model training. For example, it is much more difficult to determine which layers to freeze for more complicated object detection architectures such as Centerpoint [33].

Table 2. LiDAR segmentation ablations **with Log Mod 60 synthetic training** on the nuScenes validation dataset.

Strategy	Bicycle mIOU	Overall mIOU
Joint Training	41.2	74.8
w/ Fine-tuning	43.5	74.6
w/ Imbalance	47.6	76.6
w/ Linear Probing	48.9	76.7
w/ PolarMix	53.5	77.8

5.3. Ablating Real Data

In order to determine how effective the auxiliary domain is with less data from the original target domain, we run experiments ablating the amount of target domain data available. Specifically, we train models that have 50% and 100% of the original target domain data. We compare those results with models trained with our methodology on the auxiliary domain and these ablated datasets. In all cases, we use the same auxiliary domain dataset — the 60% log-modified dataset that performed best in the earlier experiments. The results of this experiment on the nuScenes LiDAR segmentation benchmark are shown in Figure 1. We find that as the amount of target domain data decreases, the impact of the auxiliary domain becomes more and more significant. For example, we find that using only 50% of the target domain data with all of the auxiliary domain data is sufficient to recover the full performance of training on all 100% of the target domain data.

Table 3. LiDAR segmentation ablations **with real data only** on the nuScenes validation dataset.

Strategy	Bicycle mIOU	Overall mIOU
Target-only Training	40.4	75.0
w/ Fine-tuning	40.4	75.0
w/ Imbalance	43.8	75.8
w/ Linear Probing	44.2	76.1
w/ PolarMix	45.4	76.2

5.4. Other Tasks: Object Detection

To show that we can target rare classes in settings outside of LiDAR segmentation, we use the Centerpoint [33] model to train LiDAR object detection algorithms on the same nuScenes datasets. We perform the same procedure where we simply train jointly on the two datasets and then fine-tune on the target domain only, and compare the performance with each of these auxiliary datasets to the baseline performance of training only with the nuScenes data. For this procedure, we use the class-imbalanced loss but do not add any freezing or mixing strategies. Determining exactly which layers to freeze and extending PolarMix to work well in a LiDAR object detection scenario is left to future work. However, we find that the results significantly improve even with a portion of our methodology in use. In Table 4, we see a similar story as to LiDAR segmentation. The same dataset that performed best for bicycles (Log Mod 60) also performs best for object detection. We get modest improvements on the overall NDS score with significant improvements on the bicycle class specifically.

Table 4. LiDAR object detection performance on the nuScenes validation dataset with different synthetic datasets.

Model	Bicycle AP	Overall NDS
Real Baseline	42.0	63.2
Log Mod 0	44.8	64.3
Log Mod 30	47.0	64.9
Log Mod 60	48.6	64.5
Log Mod 90	47.9	64.4

5.5. Other Classes: Debris

To validate that this methodology works on other classes as well, we use the highly challenging debris class in nuScenes, which is even more imbalanced than the bicycle class. This class is not part of the official benchmarks, since the class does not have enough data to perform well on. Indeed, the baseline model training on only nuScenes data achieves only 5 points mIoU. We perform the same strategy, replacing real traffic cones with various synthetic debris assets including trash bags, fallen signs, and construction materials in order to generate our auxiliary debris dataset. While using the auxiliary domain does not improve the debris class to something usable for autonomous driving, we find that the performance on the debris class almost doubles, as shown in Figure 7.

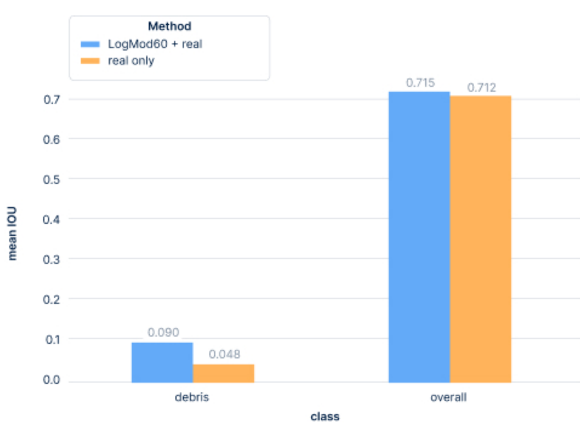


Figure 7. Performance of our methodology with a Log Mod 60 generated dataset on the debris class for LiDAR segmentation. Performance almost doubles with the addition of synthetic data.

6. Conclusion

While domain adaptation and class-imbalanced learning has been increasingly popular, there is little work using different domains to improve performance on a specific rare class in a target dataset. Further, most of the results that currently exist show these techniques for classification only, and it is unclear how well these methods may extend to object detection and segmentation. We show that straightforward modifications of new techniques in group-invariant learning can improve performance on rare classes, specifically in the LiDAR segmentation case. We significantly improve the best Cylinder3D model on the nuScenes LiDAR segmentation validation dataset and show that these techniques are even more effective when there is limited real data. Further, we show how to target specific rare classes by modifying synthetic datasets and slightly changing the loss function. This methodology works on other tasks, such as object detection, and improves performance significantly even without any of the extra techniques we have used on segmentation.

While our results are promising, there is still a significant amount of limitations in our study. First, we have shown results on a limited amount of datasets. It is unclear how well they will generalize to different domain shifts, such as SemanticKITTI to KITTI. Second, we have focused on the domain where the classes are rare but exist in the dataset. We do not consider the case where there is no target domain data of the rare class present. Finally, we have specifically tested the car-to-bicycle replacement strategy in this work. Future work could explore which classes offer a good replacement, and whether we can upsample the amount of a rare class without replacing existing classes, even though this would not allow us to match behaviors effectively. This is a particularly challenging case as it precludes the fine-tuning strategy we use here and will likely require more advanced domain adaptation. Finally, we have tested our methods purely on the LiDAR tasks for segmentation and object detection. Future work would be needed to validate these results on 2D datasets.

These limitations provide ample opportunity for future work. Other future research includes identifying better ways to generate the auxiliary domain, using auxiliary domains from other real datasets, constructing more advanced fine-tuning methods to improve rare classes, determining how to integrate more advanced partial domain adaptation methods, and using partially labeled data from the target domain.

Even with these limitations, we find the results of this study to be very relevant to constructing strong perception algorithms for autonomous driving. With this work, we have shown concrete evidence that new research in classification can be extended to real-world perception tasks, and that we can use simulated domains to target specific rare classes in these domains.

References

[1] Applied Intuition. Synthetic datasets. 3

[2] Eryk Banatt, Vickram Rajendran, and Liam Packer. Target domain data induces negative transfer in mixed domain training with disjoint classes. *arXiv preprint arXiv:2303.01003*, 2023. 1, 4

[3] Sara Beery, Yang Liu, Dan Morris, Jim Piavis, Ashish Kapoor, Neel Joshi, Markus Meister, and Pietro Perona. Synthetic examples improve generalization for rare classes. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 863–873, 2020. 2

[4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. A dataset for semantic segmentation of point cloud sequences. *CoRR*, abs/1904.01416, 2019. 3

[5] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4413–4421, 2018. 5

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1, 6

[7] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 135–150, 2018. 2

[8] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 1, 2

[9] Xiaoyu Chen, Jiachen Hu, Chi Jin, Lihong Li, and Liwei Wang. Understanding domain randomization for sim-to-real transfer. In *International Conference on Learning Representations*, 2021. 4

[10] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. `https://github.com/open-mmlab/mmdetection3d`, 2020. 5

[11] Tuhin Das, Robert-Jan Bruintjes, Attila Lengyel, Jan van Gemert, and Sara Beery. Domain adaptation for rare classes augmented with synthetic samples. *arXiv preprint arXiv:2110.12216*, 2021. 2

[12] W. Fong, R. Mohan, J. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. In *ICRA*, 2022. 3

[13] Irena Gao, Shiori Sagawa, Pang Wei Koh, Tatsunori Hashimoto, and Percy Liang. Out-of-distribution robustness via targeted augmentations. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. 2

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 1

[15] Frederik Hasecke, Pascal Colling, and Anton Kummert. Fake it, mix it, segment it: Bridging the domain gap between lidar sensors. *arXiv preprint arXiv:2212.09517*, 2022. 4

[16] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020. 4

[17] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Machine Learning*, 2022. 4, 5

[18] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022. 4

[19] Mingchen Li, Xuechen Zhang, Christos Thrampoulidis, Jiasi Chen, and Samet Oymak. Autobalance: Optimized loss functions for imbalanced data. *Advances in Neural Information Processing Systems*, 34:3163–3177, 2021. 1

[20] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018. 2

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. 5

[22] Giulio Mattolin, Luca Zanella, Elisa Ricci, and Yiming Wang. Confmix: Unsupervised domain adaptation for object detection via confidence-based mixing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 423–433, 2023. 4

[23] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 891–898, 2014. 2

[24] Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. Imbalance problems in object detection: A review. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3388–3415, 2020. 1

[25] Poojan Oza, Vishwanath A Sindagi, Vibashan VS, and Vishal M Patel. Unsupervised domain adaptation of object detectors: A survey. *arXiv preprint arXiv:2105.13502*, 2021. 2

[26] Daniel Pototzky, Azhar Sultan, and Lars Schmidt-Thieme. Parting with illusions about synthetic data. In *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, pages 1–4, 2022. 2

[27] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019. 2

[28] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European con-*

ference on computer vision, pages 443–450. Springer, 2016. 2

[29] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 4

[30] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[31] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020. 2

[32] Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. Polarmix: A general data augmentation technique for lidar point clouds. In *Advances in Neural Information Processing Systems*, 2022. 4

[33] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021. 5, 7

[34] Youshan Zhang. A survey of unsupervised domain adaptation for visual recognition. *arXiv preprint arXiv:2112.06745*, 2021. 2

[35] Qianyu Zhou, Zhengyang Feng, Qiqi Gu, Jiangmiao Pang, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Context-aware mixup for domain adaptive semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022. 4

[36] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948, 2021. 5