

Friedrich Fraundorfer
Thomas Pock
Horst Possegger (eds.)

Proceedings of the 28th Computer Vision Winter Workshop






Graz, Austria
February, 12–14, 2025

Imprint

Proceedings of the 28th Computer Vision Winter Workshop
February 12–14, 2025
Graz, Austria

Editors

Friedrich Fraundorfer	 orcid.org/0000-0002-5805-8892
Thomas Pock	 orcid.org/0000-0001-6120-1058
Horst Possegger	 orcid.org/0000-0002-5427-9938

Layout Proceedings & Cover Design


Horst Possegger
Institute of Visual Computing
Graz University of Technology

The cover image, inspired by Graz’s Old Town and Clock Tower, was generated on January 7, 2025, using [Illusion Diffusion HQ](#), a [QR code conditioned ControlNet](#) for [Stable Diffusion 1.5](#). All used generative models were released under the [CreativeML Open RAIL++-M](#) license.

ISBN 978-3-99161-022-9

DOI [10.3217/978-3-99161-022-9](https://doi.org/10.3217/978-3-99161-022-9)

2025 Verlag der Technischen Universität Graz
<https://www.tugraz-verlag.at>

This work is licensed under the  [Creative Commons Attribution 4.0 International \(CC BY 4.0\)](#) license. This Creative Commons license does not apply to third-party materials attributed to other sources or to content explicitly identified as excluded.

Contents

Preface	ii
Sponsors	ii
Workshop Organization	iii
Program Committee	iv
Author Index	v
Speakers	1
Keynote: Translating Diffusion Image Models to Other Modalities	
<i>B. Ommer</i>	2
Invited Presentations	3
Contributed Papers	4
A Data-Centric Approach to 3D Semantic Segmentation of Railway Scenes	
<i>N. Munger, M. P. Ronecker, X. Diaz, M. Karner, D. Watzenig, and J. Skaloud</i>	5
An Investigation of Beam Density on LiDAR Object Detection Performance	
<i>C. Griesbacher and C. Fruhwirth-Reisinger</i>	14
<i>Supplementary Material for An Investigation of Beam Density on LiDAR</i>	
Object Detection Performance	24
Human Pose-Constrained UV Map Estimation	
<i>M. Suchanek, M. Purkrabek, and J. Matas</i>	27
<i>Supplementary Material for Human Pose-Constrained UV Map Estimation</i>	37
Incremental Learning with Repetition via Pseudo-Feature Alignment	
<i>B. Tscheschner, E. Veas, and M. Masana</i>	38
<i>Supplementary Material for Incremental Learning with Repetition via</i>	
Pseudo-Feature Alignment	48
Leveraging Intermediate Representations for Better Out-of-Distribution Detection	
<i>G. Guglielmo and M. Masana</i>	53
Real-time Object Detection in Diverse Weather Conditions through Adaptive	
Model Selection on Embedded Devices	
<i>M. M. Tufan, C. Fruhwirth-Reisinger, M. J. Mirza, and D. ˘Stern</i>	62

Preface

Dear Colleagues,

Welcome to the **28th Computer Vision Winter Workshop (CVWW 2025)**. This year, the workshop is organized by the Institute of Visual Computing (IVC) at Graz University of Technology, and takes place in Graz, Austria, from February 12 to 14, 2025. The Computer Vision Winter Workshop is an annual international event supported by leading research groups from Ljubljana, Prague, Vienna, and Graz. It serves as a platform for researchers and PhD students to connect, exchange ideas, and foster collaboration, driving innovation in the field of computer vision. Topics of interest include image analysis, 3D vision, biometrics, human-computer interaction, vision for robotics, machine learning, and applied computer vision, among others.

This year, we received 29 submissions from various countries and institutions, including 10 contributed papers. The selection process, overseen by the Chairs, involved a rigorous double-blind review conducted by the Program Committee, comprising 40 esteemed experts in computer vision and machine learning. Each submission was reviewed by three experts, who provided detailed feedback on the strengths and weaknesses of the papers to ensure a fair and thorough evaluation. As a result of this process, 6 original contributed papers were accepted for publication and presented at oral sessions in the workshop. In addition to the contributed presentations, we are honored to host 17 invited talks featuring insights from both seasoned and early-career researchers. These were carefully selected by the Chairs in consultation with the Program Committee. A highlight of this year's program is the keynote by Prof. Björn Ommer from Ludwig Maximilian University of Munich.

We would like to express our deepest gratitude to the reviewers for their meticulous and high-quality feedback, which provided valuable insights to the authors and contributed significantly to the success of CVWW 2025. We extend our heartfelt thanks to Prof. Björn Ommer for his keynote talk. Our gratitude also goes to the mayor of the city of Graz for her sponsorship. Additionally, we are pleased to highlight outstanding work through an award sponsored by the Faculty of Computer Science and Biomedical Engineering (CSBME) at Graz University of Technology.

We hope the 28th edition of the Computer Vision Winter Workshop will be a productive and enjoyable event, sparking new ideas and fostering meaningful collaborations. Thank you for joining us!

Friedrich Fraundorfer, Thomas Pock, and Horst Possegger

CVWW General Chairs 2025

Official Sponsors

We gratefully acknowledge the support of our partners:



City of Graz















Faculty of Computer Science and
Biomedical Engineering (CSBME)

Workshop Organization

The 28th Computer Vision Winter Workshop, held in Graz from February 12–14, 2025, was organized by the **Institute of Visual Computing (IVC)**—formerly known as the **Institute of Computer Graphics and Vision (ICG)**—at Graz University of Technology. The workshop’s topics of interest encompassed a wide range of areas in computer vision, including but not limited to:

- Pattern Recognition
- Computer Vision
- Deep Learning
- Object Detection and Recognition
- Object Categorization
- 3D Vision, Stereo, and Structure from Motion
- Scene Modeling and Understanding
- Image and Video Retrieval
- Video Analysis and Event Recognition
- Statistical Methods and Learning
- Motion and Tracking
- Cognitive Vision
- Biometrics
- Face and Gesture Analysis
- Medical Image Processing
- Performance Evaluation
- Safety and Security
- Embedded Computer Vision

General Chairs

Friedrich Fraundorfer	   
Thomas Pock	   
Horst Possegger	   

Workshop Administration

Horst Possegger

Financial Administration

Charlotte Mayer
Horst Possegger



Program Committee

The Program Committee for CVWW 2025 comprised 40 esteemed experts in computer vision and machine learning. Their valuable feedback contributed significantly to the success of the workshop. The Conference Chairs are grateful for the meticulous and high-quality feedback, provided by all members of the Program Committee:

Csaba Beleznai	Austrian Institute of Technology
Verena Widhalm	Austrian Institute of Technology
Jan Čech	Czech Technical University
Ondřej Chum	Czech Technical University
Pavel Krsek	Czech Technical University
Jiří Matas	Czech Technical University
Oleksandr Shekhovtsov	Czech Technical University
Siniša Šteković	ENPC ParisTech
Dániel Baráth	ETH Zurich
Levente Hajder	Eötvös Loránd University
Christian Fruhwirth-Reisinger	Graz University of Technology
Robert Harb	Graz University of Technology
Georg Krispel	Graz University of Technology
Dušan Malić	Graz University of Technology
Marc Masana	Graz University of Technology
Jakub Micorek	Graz University of Technology
Lukas Radl	Graz University of Technology
David Schinagl	Graz University of Technology
Michael Steiner	Graz University of Technology
Martin Zach	Graz University of Technology
Jun Zhang	Graz University of Technology
Roland Perko	Joanneum Research
Samuel Schulter	NEC Laboratories America
Luka Čehovin Zajc	University of Ljubljana
Matej Dobrevski	University of Ljubljana
Žiga Emeršič	University of Ljubljana
Matej Kristan	University of Ljubljana
Alan Lukežič	University of Ljubljana
Luka Šajn	University of Ljubljana
Domen Tabernik	University of Ljubljana
Peter M. Roth	University of Veterinary Medicine Vienna
Lea Bogensperger	University of Zurich
Margrit Gelautz	Vienna University of Technology
Pedro Hermosilla Casajus	Vienna University of Technology
Martin Kampel	Vienna University of Technology
Florian Kleber	Vienna University of Technology
Andreas Kriegler	Vienna University of Technology
Robert Sablatnig	Vienna University of Technology
Markus Vincze	Vienna University of Technology
Sebastian Zambanini	Vienna University of Technology

Author Index

D

Diaz, Xavier 5

F

Fruhworth-Reisinger, Christian 14, 62

G

Griesbacher, Christoph 14

Guglielmo, Gianluca 53

K

Karner, Michael 5

M

Masana, Marc 38, 53

Matas, Jiří 27

Mirza, Muhammad Jehanzeb 62

Münger, Nicolas 5

P

Purkrábek, Miroslav 27

R

Ronecker, Max Peter 5

S

Skaloud, Jan 5

Štern, Darko 62

Suchánek, Matěj 27

T

Tscheschner, Benedikt 38

Tufan, Mohammad Milad 62

V

Veas, Eduardo 38

W

Watzenig, Daniel 5

Speakers



Keynote Talk

Translating Diffusion Image Models to Other Modalities

Prof. Björn Ommer

Ludwig Maximilian University of Munich

Recently, generative models for learning image representations have seen unprecedented progress. Approaches such as diffusion models and transformers have been widely adopted for various tasks related to visual synthesis, modification, analysis, retrieval, and beyond. Despite their enormous potential, current generative approaches have their own specific limitations. We will discuss how recently popular strategies such as flow matching can significantly enhance efficiency and democratize AI by empowering smaller models. The main part of the talk will then investigate effective ways to utilize pretrained diffusion-based image synthesis models for different tasks and modalities. Therefore, we will efficiently translate powerful generative image representations to different modalities and show evaluations on other tasks.

Short Speaker Biography

Prof. Björn Ommer is a full professor at the Ludwig Maximilian University of Munich (LMU) where he heads the Computer Vision & Learning (CompVis) Group. Previously, he was a full professor in the Department of Mathematics and Computer Science of Heidelberg University. At Heidelberg, he also served as one of the directors of the Interdisciplinary Center for Scientific Computing (IWR) and of the Heidelberg Collaboratory for Image Processing (HCI). The CompVis Group focuses on fundamental research in computer vision and machine learning, with applications spanning diverse fields such as the digital humanities and life sciences.

Björn Ommer studied computer science with a minor in physics at the University of Bonn. He earned his Ph.D. in computer science from ETH Zurich, where his dissertation, "Learning the Compositional Nature of Objects for Visual Recognition", was honored with the ETH Medal. Following this, he worked as a postdoctoral researcher in Jitendra Malik's Computer Vision Group at UC Berkeley.

He is a member of the Bavarian AI Council and serves as an associate editor for IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), having previously held the same role for Pattern Recognition Letters. Björn is an ELLIS Fellow, faculty member of the ELLIS Unit Munich, affiliated with the Helmholtz Foundation, and a principal investigator at the Munich Center for Machine Learning (MCML). He has held prominent roles at leading conferences, serving as Program Chair for GCPR, Senior Area Chair and Area Chair for CVPR, ICCV, ECCV, and NeurIPS, and as a workshop and tutorial organizer at these venues. In 2023, Björn delivered the opening keynote at NeurIPS. His work on Stable Diffusion has been nominated for the German Future Prize of the President of Germany, and in 2024, he was awarded the German AI Prize.

Invited Presentations

CVWW 2025 hosted 17 invited talks featuring insights from both seasoned and early-career researchers. The following speakers were carefully selected by the Chairs in consultation with the Program Committee:

Klára Janoušková	Czech Technical University in Prague
Miroslav Purkrábek	Czech Technical University in Prague
Jan Škvrna	Czech Technical University in Prague
Alena Smutná	Czech Technical University in Prague
Levente Hajder	Eötvös Loránd University
Christian Fruhwirth-Reisinger	Graz University of Technology
Dušan Malić	Graz University of Technology
Lukas Radl	Graz University of Technology
Matic Fučka	University of Ljubljana
Blaž Rolih	University of Ljubljana
Peter Rot	University of Ljubljana
Leon Todorov	University of Ljubljana
Jovana Videnović	University of Ljubljana
Filip Wolf	University of Ljubljana
Anja Delić	University of Zagreb
Ivan Martinović	University of Zagreb
Tingyu Lin	Vienna University of Technology

Contributed Papers



A Data-Centric Approach to 3D Semantic Segmentation of Railway Scenes

Nicolas Münger Max Ronecker Xavier Diaz Michael Karner

SETLabs Research GmbH
Elsenheimerstraße 55, 80687 München, Germany
`{firstname.lastname}@setlabs.de`

Daniel Watzenig
Graz University of Technology
Inffeldgasse 16/II, 8010 Graz, Austria
`daniel.watzenig@tugraz.at`

Jan Skaloud
EPFL - Swiss Federal Technology Institute of Lausanne
GR A2 392 (Bâtiment GR) , 1015 Lausanne , Switzerland
`jan.skaloud@epfl.ch`

Abstract

LiDAR-based semantic segmentation is critical for autonomous trains, requiring accurate predictions across varying distances. This paper introduces two targeted data augmentation methods designed to improve segmentation performance on the railway-specific OSDaR23 dataset. The person instance pasting method enhances segmentation of pedestrians at distant ranges by injecting realistic variations into the dataset. The track sparsification method redistributes point density in LiDAR scans, improving track segmentation at far distances with minimal impact on close-range accuracy. Both methods are evaluated using a state-of-the-art 3D semantic segmentation network, demonstrating significant improvements in distant-range performance while maintaining robustness in close-range predictions. We establish the first 3D semantic segmentation benchmark for OSDaR23, demonstrating the potential of data-centric approaches to address railway-specific challenges in autonomous train perception.

1. Introduction

Rail transport offers a sustainable alternative to other transportation modes, emitting significantly lower carbon emissions [11]. Its continued development, especially through autonomous train operation (ATO), is critical to achieving climate goals like those in the European Union’s Green Deal. ATO, defined from GoA0 (manual) to GoA4 (fully automated) [24], addresses labor shortages, increases operational flexibility and reliability, and optimizes service frequency. The Lausanne metro M2

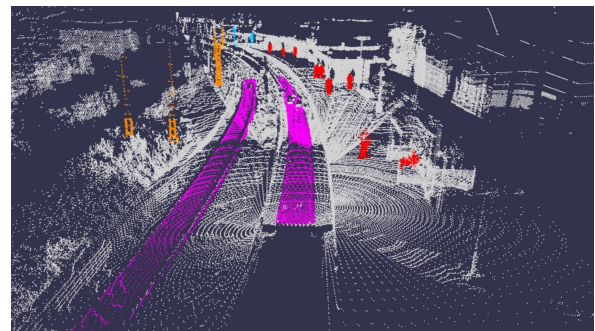


Figure 1. Example of a segmented pointcloud from the OSDaR23 dataset [30]

line, a GoA4 system, demonstrates these benefits through higher frequency and adaptability. However, while fully automated systems work well in controlled settings, such as metro lines, implementing GoA3–4 in open rail networks is challenging due to unpredictable obstacles and the absence of physical barriers. Ensuring safety in open rail ATO is therefore a key research area.

Robust perception systems are essential for obstacle detection and hazard identification in ATO. LiDAR (Light Detection and Ranging) suits these tasks by providing rich 3D geometric information [21]. LiDAR semantic segmentation, assigning a class to each 3D point, enables detailed environmental understanding. In autonomous driving, 3D object detection [6, 7, 18, 23, 25, 26] and semantic segmentation [1, 17, 19, 27, 38] are well-studied across many modalities. However, applying these techniques to autonomous train operation has received less attention, partly due to limited public datasets. The OSDaR23 dataset [30] addresses this gap by providing data

for various railway perception tasks (Fig. 1). This paper applies deep learning-based 3D semantic segmentation to LiDAR point clouds in the railway domain using OSDaR23. We focus on safety-critical classes, emphasizing long-range segmentation accuracy due to trains’ substantial braking distances. We also adopt a data-centric approach, introducing domain-specific data augmentations to improve robustness and performance.

Contributions

This paper introduces targeted data augmentation methods for LiDAR semantic segmentation in the railway domain, evaluated on the real-world OSDaR23 dataset.

1. Comprehensive evaluation of a state-of-the-art 3D semantic segmentation network on OSDaR23, including dataset analysis.
2. A person instance pasting augmentation method to enhance pedestrian segmentation at distant ranges.
3. A track sparsification augmentation method to improve track segmentation by redistributing point density.
4. Report the first 3D semantic segmentation results on the OSDaR23 dataset.

2. Background

This background section provides a general overview of point cloud segmentation, followed by segmentation and augmentation techniques specific to the railway domain.

2.1. Point cloud semantic segmentation

Semantic segmentation assigns a class label to each element of the input. While image-based segmentation assigns labels to pixels, point cloud segmentation must handle unordered, unstructured 3D points. Deep learning has become the standard approach, surpassing traditional techniques [35]. Methods are typically categorized into view-based, voxel-based, and point-based approaches, each imposing structure onto the raw data differently.

View-based methods

View-based methods project the point cloud into one or multiple 2D images, leveraging established image-based segmentation. SnapNet [4] generates RGB-depth snapshots from various viewpoints, applies a CNN for labeling, and back-projects labels to 3D. CENet [8] uses spherical projection and channels (x, y, z, d, r) for each pixel. Larger image widths improve performance but slow inference. However, these methods lose some 3D geometric fidelity due to projection.

Voxel-based methods

Voxel-based methods discretize the point cloud into a volumetric grid and apply 3D CNNs. PVKD [14], for example, builds on Cylinder3D [41] and employs a teacher-student framework, achieving similar accuracy at lower latency. Despite structuring the data, voxelization introduces resolution limits and can demand high memory.

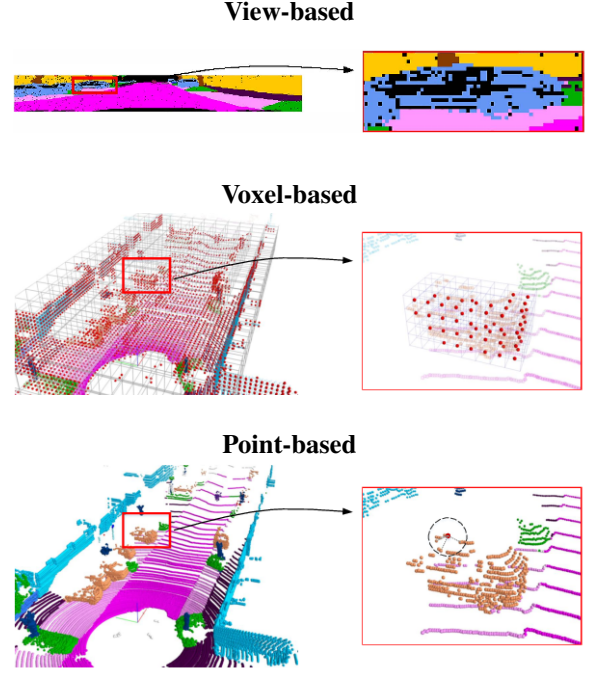


Figure 2. Schematic representation of three main deep learning-based methods for semantic segmentation of point cloud data. Adapted from [36].

Point-based methods

Point-based methods directly process points without explicit restructuring. PointNet [21] introduced MLP-based features and max-pooling for permutation invariance. Transformers, as in Point Transformer [39] and its improved PTV3 [33], leverage self-attention for robust performance. This preserves data fidelity but can be slower.

In summary, view-based and voxel-based methods effectively impose structure at the cost of fidelity, while point-based methods maintain full data integrity but may be computationally more demanding.

Fig. 2 shows an example for each of the three approaches.

2.2. Railway-domain focused segmentation

Prior work on railway point cloud segmentation focused mainly on infrastructure inspection. [28] segmented tunnel scenes into ground, lining, wiring, and rails using KP-Conv [31] and PointNet [21]. Similarly, [13] employed a PointNet++ [22]-based architecture to classify rails, cables, and traffic signals. These efforts used non-public datasets and older architectures, and did not target autonomous train operation.

In contrast, the automotive field has benefited from large-scale, publicly available datasets like Waymo Open Dataset [29], nuScenes [5], and SemanticKITTI [2]. Comparable resources remain scarce in the railway domain. Existing sets, such as WHU-Railway3D [12] and Rail3D [15], focus on infrastructure and rely on multi-frame reconstructions, not reflecting real-time conditions. OSDaR23 [30] addresses this gap with single-frame Li-

DAR data and classes relevant to autonomous rail operation, enabling models tailored to open railway environments.

2.3. Data augmentation methods for point clouds

Data-centric AI aims to enhance model performance by improving data quality and diversity rather than solely refining architectures. In point cloud segmentation, data augmentation (DA) introduces variations—such as rotations, translations, and sparsifications—to enrich training data and improve generalization [9, 20, 40].

Part-aware augmentation [10] applies transformations to specific object regions (e.g., sparsifying parts of cars or pedestrians), reducing reliance on dense shapes and aiding recognition at longer distances. PolarMix [34] integrates entire LiDAR scans by angular swapping or instance-level rotate-pasting, increasing variability at both scene and object levels. Both methods have demonstrated notable performance gains in 3D tasks and inspire the DA techniques explored in this work.

3. Initial Analysis

In this section, we evaluate the baseline performance of Point Transformer V3 (PTV3) on the OSDaR23 dataset. Since the dataset has seen limited use in prior research, its suitability for semantic segmentation tasks, along with potential performance bottlenecks, remains unclear. This analysis aims to establish a baseline understanding of the model’s strengths and limitations, highlighting key challenges such as class imbalance and long-range prediction issues. These findings will guide subsequent efforts to enhance model performance through targeted improvements.

3.1. Baseline

For our baseline, we require a modern, high-performing semantic segmentation model suited for LiDAR point clouds. Point Transformer V3 (PTV3)[33] is the current top performer on the SemanticKITTI benchmark, demonstrating strong segmentation accuracy with reasonable inference speed. Although relatively new and less cited, it builds on the widely adopted Point Transformer[32, 39] architecture, making it a robust choice for our experiments.

3.2. Dataset and Experiment setup

We conduct our experiments on OSDaR23 [30], a single-frame, multi-sensor LiDAR dataset collected in various railway scenarios. As shown in Table 1, OSDaR23 has a higher average point density per frame than popular automotive datasets [2, 5, 29], but covers fewer total frames and primarily captures the forward view of the locomotive instead of a full 360° surround.

Although OSDaR23 provides 22 annotated classes, several contain few points, resulting in class imbalance (Fig. 3a). To address this, we merge or discard certain classes (Table 2) and remove overlapping annotations

(e.g., *switch on track*). Figure 3b shows the resulting distribution after class mapping.

All experiments follow the official train, validation, and test splits. We adapt data augmentations to the forward-facing LiDAR viewpoint, limiting large rotations/flips and applying sensor-specific intensity normalization. We train Point Transformer V3 (PTV3) with a learning rate of 0.001, using both cross-entropy and Lovász-Softmax loss [3].

3.3. Baseline Performance

We begin by examining the baseline model’s overall segmentation performance on the validation set. As shown in Table 3, the model (PTV3) achieves a mean IoU (mIoU) of 74.49%, indicating solid overall accuracy across classes. However, this summary metric masks performance issues at longer ranges.

Fig. 4 shows the recall map for the class track. For each planar grid cell of 1x1 meter, the recall is computed. The values are obtained over all frames of the validation set, providing an overview of the performances given the spatial location. In the ranges close to the sensor the recall is generally high. Beyond x=60m, however, the recall quickly degrades. This means the network has good capabilities at identifying the track points at close range but misses points further.

Similarly, person segmentation suffers at longer ranges, as reflected in the range IoU (rIoU) results (Table 4). Although performance is strong at mid-range (40–60 m), it drops significantly beyond 60 m. This decline correlates with fewer training samples at longer distances, indicating that data scarcity limits long-range accuracy.

In summary, while the baseline model performs well overall, it struggles to maintain performance at longer distances for key classes like track and person. Insufficient training data in these ranges is a likely contributor to weaker performance, motivating the need for data augmentation and other strategies to improve long-range segmentation results.

4. Methodology

This section outlines the data-centric strategies developed to address the dataset-related limitations identified in the baseline analysis. Our methodology focuses on two key augmentations: track sparsification and person instance pasting, tailored to the characteristics of the OSDaR23 dataset.

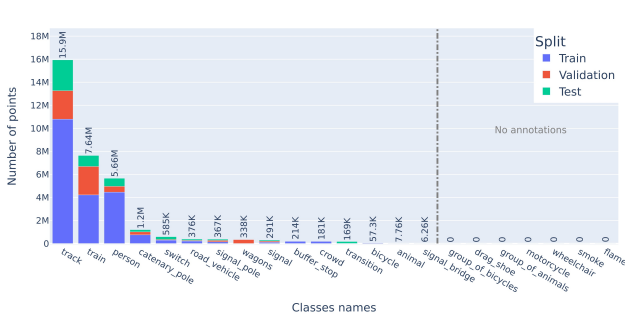
4.1. Tracks sparsification

Building on the part-aware data augmentation method [10], a new strategy was developed to improve track prediction accuracy at farther ranges.

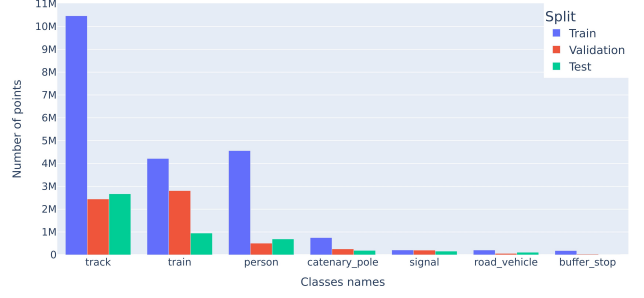
Dense parts of track instances are sparsified by adapting the number of points per range for each track instance. The goal is to equalize point density by reducing points near the sensors to match the density farther away. This is

Table 1. Comparison of OSDaR23 to other popular autonomous driving point cloud datasets.

	SemanticKITTI [2]	NuScenes [5]	Waymo [29]	OSDaR23 [30]
Avg. Points/Frame	120K	34K	177K	204K
Ann. LiDAR frames	15K	40K	230K	1.5K
# LiDAR sources	1	1	5	6
360° field of view	✓	✓	✓	✗



(a) Points per class of the OSDaR23 dataset before mapping.



(b) Points per class of the OSDaR23 dataset after mapping (background omitted).

Figure 3. Comparison of OSDaR23 class distributions before and after mapping.

Table 2. Class mapping for OSDaR23.

Original classes	Mapped class
person, crowd	person
train, wagons	train
bicycle, animal, signal_bridge	background
transition, track	track
road_vehicle	road_vehicle
catenary_pole	catenary_pole
signal_pole, signal	signal
buffer_stop	buffer_stop
switch	discarded

achieved by evaluating the number of points within a window of width W at a distance d from the origin, where C_{max} represents the point count in the farthest range. Closer ranges are then randomly downsampled to match C_{max} , ensuring uniform density.

Let $P_{track,i}[d-W,d]$ denote the set of points belonging to the i^{th} track instance in the planar distance range $[d-W, d]$. The variables W (window width) and C_{max} can

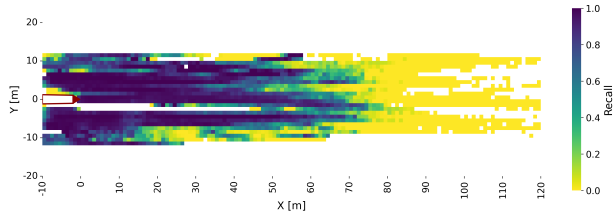


Figure 4. Recall for the class track across the validation set. High recall is observed close to the sensor, with performance decreasing beyond 60 m.

be adjusted based on sensor specifications and use case requirements. The pseudocode for the transformation is provided in Algorithm 1.

This procedure is applied to all track instances in a frame. Fig. 5 shows a point cloud before and after the transformation. In this example, the window width W is set to 10 meters, and C_{max} is set to 80 meters. The desired density is determined within the range $[C_{max} - W, C_{max}]$ (70–80 meters). Points beyond 70 meters remain unchanged, while those closer than 70 meters are significantly downsampled.

4.2. Person Instance Pasting

Inspired by PolarMix [34], we developed a methodology to paste person instances from one frame into another during training. This approach diversifies pedestrian samples by increasing their population. Unlike PolarMix, where

Algorithm 1 Track Instance Sparsification

Input: $P_{t,i}$ (points of track i), d_{max} (upper range), W (window width)
Output: Downsampled $P_{t,i}$
 $D_i \leftarrow$ planar distances from origin for $P_{t,i}$
 $d_{max} \leftarrow \min(d_{max}, \max(D_i))$
 $C_{max} \leftarrow$ count points in $[d_{max} - W, d_{max}]$
while $d_{max} > 0$ **do**
 $d_{max} \leftarrow d_{max} - W$
 $C \leftarrow$ count points in $[d_{max} - W, d_{max}]$
 if $C > C_{max}$ **then**
 Remove $C - C_{max}$ points from $P_{t,i}$
 end if
end while
Return $P_{t,i}$

Table 3. Summary results for the baseline experiment on the validation dataset.

IoU (validation set)									mIoU
background	person	train	road vehicle	track	catenary pole	signal	buffer stop	Overall	
96.84	69.65	86.39	70.09	82.89	47.40	48.80	93.86	74.49	

Table 4. Baseline range-based IoU for the person class and approximate number of training instances.

Distance range	IoU [%] (Val)	#Instances (Train)
0–20 m	80.40	≈ 5900
20–40 m	69.73	≈ 3500
40–60 m	81.23	≈ 500
60–80 m	31.36	≈ 350
80–100 m	45.17	≈ 100

objects are rotated around the vehicle without individual transformations, our method accounts for the forward-facing point clouds in OSDaR23, which differ from the 360-degree coverage in datasets like SemanticKITTI. A simple rotation would place instances outside the field of view, necessitating significant adaptation of the original methodology.

As in PolarMix, Scan *A* denotes the frame undergoing transformation, and Scan *B* denotes the randomly selected frame from the training set containing at least one person instance.

Each instance of Scan *B* goes through a set of individual transformations, applied in this order:

1. Flipping along the X axis with 0.5 probability.
2. Random rotation around the instance’s center along the Z axis, within the range $[-180^\circ, 180^\circ]$.
3. Random shift along the Y axis, within the range $[-2\text{m}, 2\text{m}]$.
4. Random shift towards the back of the scene, along the X axis.
5. Shifting along the Z axis so as to be at a realistic height.

An example for scan A and B and the produced result

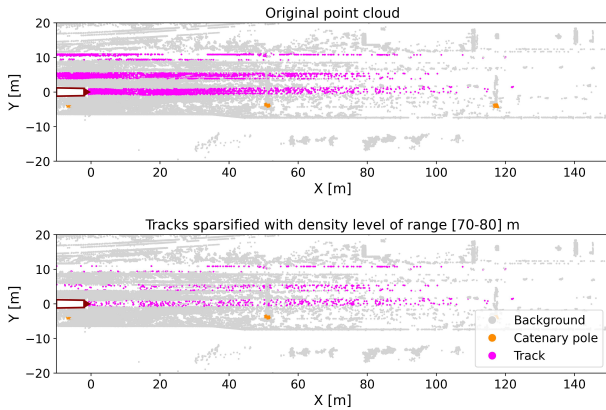


Figure 5. Effect of the tracks sparsification transformation on scene 3_fire_site_3.1, frame 58 from the OSDaR23 dataset.

is shown in Fig. 6.

For the X-axis shift, instances are translated further from the sensor to balance the distribution, with density adjustments based on the histogram of points per instance. The instance is downsampled to match the expected point count N , sampled randomly within $[N-0.1N, N+0.1N]$.

For the Z-axis shift, instances are adjusted to align with the ground. The ground height is estimated as the mean height of points in Scan *A* under the instance’s bounding box. Special cases include estimating the ground height from railway tracks when no points overlap or ignoring unrealistic heights (e.g., above 150 cm).

After applying the transformation, the augmented dataset shows a more balanced distribution of person instances across ranges, particularly in previously sparse areas as shown in Fig 7.

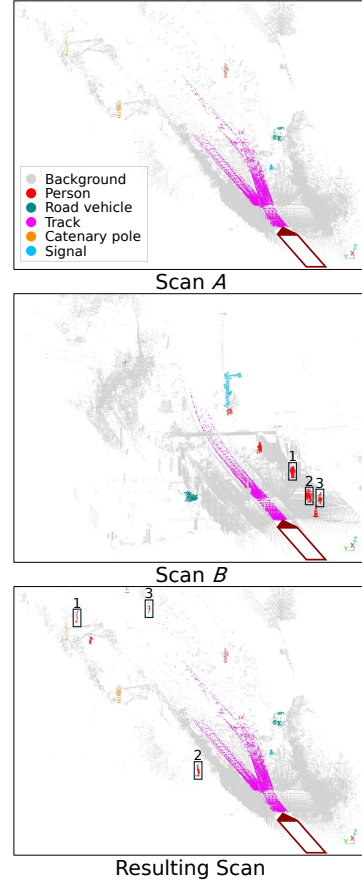


Figure 6. Visualisation of the person instances pasting transformation. Best viewed zoomed in.

5. Results

This section presents the results of applying the data augmentation (DA) methods during training, with varying proportions of affected samples. Models are first evaluated on the validation set to select the best for each task, which are then tested on the test set.

To reduce the foreground bias of IoU, we propose the mean range IoU (mean rIoU), which assigns equal importance to IoUs across all ranges. Let rIoU_i represent the range IoU for bin i . The mean rIoU is defined as:

$$\text{mean rIoU} = \frac{1}{N} \sum_{i=1}^N \text{rIoU}_i \quad (1)$$

where rIoU_i is computed for points in the range $[r_{\min,i}, r_{\max,i}]$, with $r_{\min,i}$ and $r_{\max,i}$ as bin boundaries, and N as the number of bins.

5.1. Track sparsification

This section evaluates the impact of the track sparsification DA method, tested with two density selection distances (DSD): 70-80m and 40-50m. The augmentation was applied with varying probabilities (p) during training, with range IoUs computed at 20m intervals from 0-100m. The baseline corresponds to $p = 0$ (no augmentation), while $p = 1$ applies the transformation to all training samples.

The ablation study identifies the best augmentation probabilities as $p = 0.6$ for DSD 70-80m and $p = 0.9$ for DSD 40-50m. Table 5 summarizes the results. The model with DSD 40-50m at $p = 0.9$ achieves the highest mean rIoU (59.49%), improving performance in ranges 40-60m and 60-80m by over 7 percentage points compared to the baseline. Both augmented models show improvements in the farthest range (80-100m), while maintaining strong performance near the origin. The baseline achieves the highest rIoU in 0-20m but with minimal difference (0.01 percentage points).

The selected model (DSD 40-50m, $p = 0.9$) also improves recall at farther distances, as shown in Fig. 8, while maintaining comparable performance closer to the origin. The results demonstrate that the track sparsification DA

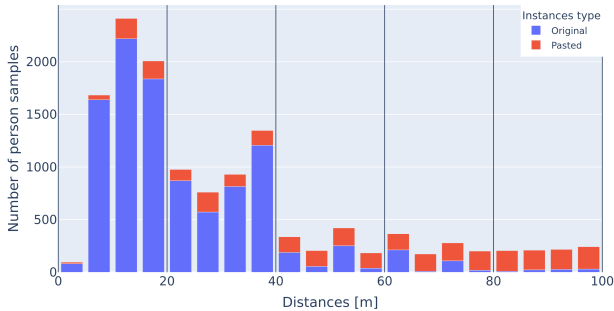


Figure 7. New distribution of samples with the person instance pasting DA applied on all frames from the train set.

method effectively enhances performance at greater distances when applied with the identified optimal probabilities.

5.2. Person instances pasting

This section evaluates the impact of the person instances pasting DA method using two approaches: online augmentation and offline dataset inflation. Online augmentation applies transformations to training samples in real-time, modifying data on-the-fly during training. Offline augmentation pre-processes the dataset by adding transformed samples, increasing its size before training. For person instance pasting, online augmentation randomly pastes instances during training, while offline augmentation generates augmented frames beforehand and incorporates them into the dataset.

In online augmentation, the probability (p) determines the likelihood of applying transformations to a sample during each training iteration. Higher p dynamically increases the number of augmented samples in each epoch.

In offline augmentation, the dataset size is expanded by adding transformed samples, controlled by the augmentation ratio (α). For instance, $\alpha = 1.0$ doubles the dataset by adding a transformed version of each sample, while $\alpha = 0.5$ increases the size by 50%.

Again an ablation study is conducted to determine the optimal values for p and α . The best models are selected based on mean rIoU: $p = 0.8$ for online DA and $\alpha = 0.1$ for offline DA. Table 5 compares these models with the baseline. Both approaches show significant improvements in the farthest ranges (60-100m). The online method achieves an 18.56 percentage-point increase in range 60-80m and a 12.59-percentage-point increase in range 80-100m over the baseline. Similarly, the offline method improves range 80-100m by 13.53 percentage-point. For closer ranges (0-60m), the differences are minimal, with variations below 3 percentage points. The online DA trained model ($p = 0.8$) achieves the highest mean rIoU (66.99%) and is the overall best model for this task.

5.3. Results on Test Set

The best-performing models identified during validation were evaluated on the test set to assess their generalization to new data.

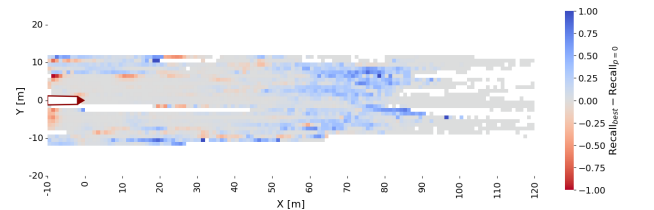


Figure 8. Recall difference between the best model and model with no augmentation on the validation set.

Table 5. Summary metrics for baseline and best models of track sparsification and person instance pasting (validation set).

	mean rIoU	r0-20	r20-40	r40-60	r60-80	r80-100
Track Sparsification (Density Selection Distances)						
baseline	56.52	86.76	82.05	64.98	40.98	7.82
70-80m (best)	58.01	86.64	81.61	64.74	43.15	13.93
40-50m (best)	59.49	86.75	82.19	66.70	48.29	13.50
Person Instances Pasting						
baseline	61.57	80.40	69.73	81.23	31.36	45.17
online (best)	66.99	78.66	70.12	78.49	49.92	57.76
offline (best)	66.77	80.98	70.12	79.46	44.59	58.70

Table 6. Summary of test set results. TS: track sparsification, PIP: person instance pasting (online). For each method, the best-performing model from the validation set is used.

	IoU (test set)								mIoU
	background	person	train	road vehicle	track	catenary pole	signal	buffer stop	
Baseline	97.09	77.98	59.87	72.06	81.29	71.01	56.83	0.53	64.58
TS (best)	97.03	77.27	57.33	73.51	80.60	75.67	53.27	0.29	64.37
PIP online (best)	97.02	77.21	57.47	77.33	81.34	75.83	52.25	0.81	64.91

5.3.1. Class Track

The best model for track sparsification (TS, DSD 40-50m, $p = 0.9$) improves rIoUs in ranges beyond 40m, with a 5 percentage-point increase in 80-100m compared to the baseline. However, a slight decrease in the 0-20m range is observed, attributed to the network focusing on sparsified far-range points during training, potentially neglecting the dense close-range regions. Recall maps show significant gains in 60-90m, reflecting better far-range detection, while closer ranges see some localized recall reduction on the locomotive’s sides.

5.3.2. Class Person

The best model for person instance pasting (PIP online, $p = 0.8$) achieves substantial improvements in distant ranges, with increases of 11.42 and 12.59 percentage points in 60-80m and 80-100m, respectively. However, a drop of 11.58 points in the 40-60m range is linked to low diversity in the test set for this range, dominated by repetitive samples of a single stationary human instance. These repetitive samples, while well-segmented across frames, contribute to cumulative small errors, reducing the rIoU.

5.3.3. Other Classes

Table 6 summarizes the IoUs across all classes. The baseline model performs best overall for the person class, while the PIP online model achieves the highest track IoU. These results highlight that the methods are tailored to improve distant-range performance, leading to trade-offs in close-range inference. For the buffer stop class, all models show a near-complete IoU drop (from 93.86% on validation to <1% on the test set), due to overfitting to similar training-validation point clouds and poor generalization to the sparse test set.

5.3.4. Discussion of Results

The TS method enhances far-range performance while minimally impacting close-range inference, demonstrating its effectiveness in handling sparsified regions. Future work could explore variable DSDs for improved adaptability.

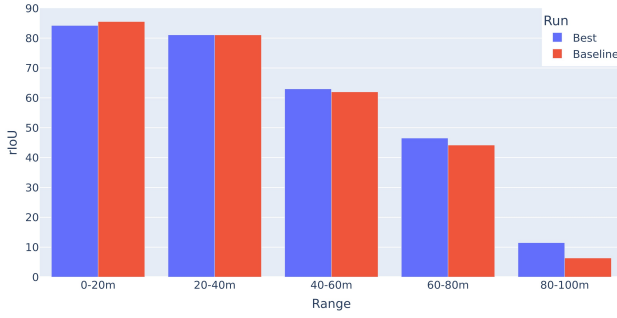
The PIP online method significantly boosts distant-range rIoUs but struggles in low-diversity regions such as 40-60m. Future improvements could include adapting the intensity field and creating a more diverse instance registry to enhance generalization.

6. Conclusion

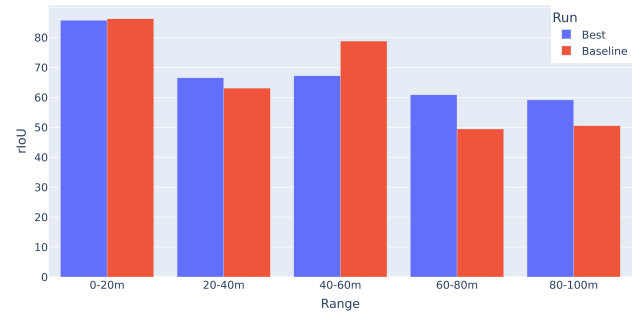
The experiments on OSDaR23 validate the effectiveness of the proposed targeted data augmentations in improving segmentation performance at distant ranges, with minimal impact on close-range accuracy. The track sparsification and person instance pasting methods address key challenges in LiDAR-based semantic segmentation for autonomous trains.

Future work could integrate additional sensor data, such as RGB images, to leverage color information and enhance performance. Incorporating temporal data, as demonstrated in methods like MemorySeg [16], could further improve predictions by capturing motion and context. Additionally, exploring the inverse of track sparsification—densifying distant point clouds using techniques like [37]—offers another avenue for enhancing segmentation in sparse regions.

These methods provide a solid foundation for advancing multimodal, temporal, and augmentation-driven approaches in semantic segmentation for autonomous train systems.



(a) Class track: Range IoUs for baseline and TS model.



(b) Class person: Range IoUs for baseline and PIP model.

Figure 9. Comparison of range IoUs on the test set for baseline and the best-performing models: (a) Track sparsification (TS), (b) Person instance pasting (PIP online).

Acknowledgements

This work has received funding from the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under grant agreement 19I21039A.

References

- [1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017. 1
- [2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 2, 3, 4
- [3] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, Salt Lake City, UT, 2018. IEEE. 3
- [4] Alexandre Boulch, Bertrand Le Saux, and Nicolas Audebert. Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks, 2017. 2
- [5] 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, 2020. 2, 3, 4
- [6] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, Los Alamitos, CA, USA, 2017. IEEE Computer Society. 1
- [7] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1
- [8] Hui-Xian Cheng, Xian-Feng Han, and Guo-Qiang Xiao. Cenet: Toward Concise and Efficient Lidar Semantic Segmentation for Autonomous Driving. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 01–06, 2022. 2
- [9] Shuyang Cheng, Zhaoqi Leng, Ekin Dogus Cubuk, Barret Zoph, Chunyan Bai, Jiquan Ngiam, Yang Song, Benjamin Caine, Vijay Vasudevan, Congcong Li, Quoc V. Le, Jonathon Shlens, and Dragomir Anguelov. Improving 3D Object Detection through Progressive Population Based Augmentation, 2020. 3
- [10] Jaeseok Choi, Yeji Song, and Nojun Kwak. Part-Aware Data Augmentation for 3D Object Detection in Point Cloud, 2021. 3
- [11] Deutsche Bahn. Metropolitan Network: A strong European railway for an ever closer union, 2023. 1
- [12] Zhen Dong, Fuxun Liang, Bisheng Yang, Yusheng Xu, Yufu Zang, Jianping Li, Yuan Wang, Wenxia Dai, Hongchao Fan, Juha Hyypä, and Uwe Stilla. Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 163:327–342, 2020. 2
- [13] Javier Grandio, Belén Riveiro, Mario Soilán, and Pedro Arias. Point cloud semantic segmentation of complex railway environments using deep learning. *Automation in Construction*, 141:104425, 2022. 2
- [14] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-Voxel Knowledge Distillation for LiDAR Semantic Segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8469–8478, New Orleans, LA, USA, 2022. IEEE. 2
- [15] Abderrazzaq Kharroubi, Zouhair Ballouch, Rafika Hajji, Anass Yarroudh, and Roland Billen. Multi-Context Point Cloud Dataset and Machine Learning for Railway Semantic Segmentation. *Infrastructures*, 9(4):71, 2024. 2
- [16] Enxu Li, Sergio Casas, and Raquel Urtasun. MemorySeg: Online LiDAR Semantic Segmentation with a Latent Memory. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 745–754, Paris, France, 2023. IEEE. 7
- [17] Yanwei Li, Xinze Chen, Zheng Zhu, Lingxi Xie, Guan Huang, Dalong Du, and Xingang Wang. Attention-Guided Unified Network for Panoptic Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7019–7028, Los Alamitos, CA, USA, 2019. IEEE Computer Society. 1
- [18] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfu-

- sion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 1
- [19] Rohit Mohan and Abhinav Valada. Efficienttps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129:1551–1579, 2020. 1
- [20] Jiquan Ngiam, Benjamin Caine, Wei Han, Brandon Yang, Yuning Chai, Pei Sun, Yin Zhou, Xi Yi, Ouais Alsharif, Patrick Nguyen, Zhifeng Chen, Jonathon Shlens, and Vijay Vasudevan. StarNet: Targeted Computation for Object Detection in Point Clouds, 2019. 3
- [21] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, 2017. 1, 2
- [22] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2
- [23] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum PointNets for 3D Object Detection from RGB-D Data. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 918–927, Los Alamitos, CA, USA, 2018. IEEE Computer Society. 1
- [24] Giuseppe Rizzi. Automated metros. *UITP*, Accessed: 5 Sept. 2024. 1
- [25] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointnet: 3d object proposal generation and detection from point cloud. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1
- [26] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Peter Kotschieder, and Elisa Ricci. Are we missing confidence in pseudo-lidar methods for monocular 3d object detection? In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3205–3213, 2021. 1
- [27] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. Efficienttps: Efficient lidar panoptic segmentation. *IEEE Transactions on Robotics*, 38(3):1894–1914, 2022. 1
- [28] M. Soilán, A. Nóvoa, A. Sánchez-Rodríguez, B. Riveiro, and P. Arias. Semantic segmentation of point clouds with pointnet and kpconv architectures applied to railway tunnels. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2020:281–288, 2020. 2
- [29] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset, 2020. 2, 3, 4
- [30] Rustam Tagiev, Martin Köppel, Karsten Schwalbe, Patrick Denzler, Philipp Neumaier, Tobias Klockau, Martin Boekhoff, Pavel Klasek, and Roman Tilly. OSDaR23: Open Sensor Data for Rail 2023. In *2023 8th International Conference on Robotics and Automation Engineering (ICRAE)*, pages 270–276, 2023. 1, 2, 3, 4
- [31] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotequi, Francois Goulette, and Leonidas J. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019. 2
- [32] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point Transformer V2: Grouped Vector Attention and Partition-based Pooling, 2022. 3
- [33] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point Transformer V3: Simpler, Faster, Stronger, 2023. 2, 3
- [34] Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. PolarMix: A General Data Augmentation Technique for LiDAR Point Clouds. 2022. 3, 4
- [35] Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu. Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):38–59, 2020. 2
- [36] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16004–16013, Montreal, QC, Canada, 2021. IEEE. 2
- [37] Jihwan You and Young-Keun Kim. Up-Sampling Method for Low-Resolution LiDAR Point Cloud to Enhance 3D Object Detection in an Autonomous Driving Environment. *Sensors*, 23(1):322, 2023. 7
- [38] Jiaying Zhang, Xiaoli Zhao, Zheng Chen, and Zhejun Lu. A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. *IEEE Access*, 7:179118–179133, 2019. 1
- [39] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 2, 3
- [40] Qinfeng Zhu, Lei Fan, and Ningxin Weng. Advancements in Point Cloud Data Augmentation for Deep Learning: A Survey. *Pattern Recognition*, 153:110532, 2024. 3
- [41] 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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9934–9943, Nashville, TN, USA, 2021. IEEE. 2

An Investigation of Beam Density on LiDAR Object Detection Performance

Christoph Griesbacher¹

Christian Fruhwirth-Reisinger^{1,2}

¹Institute of Visual Computing, TU Graz

²Christian Doppler Laboratory for Embedded Machine Learning

{griesbacher, reisinger}@tugraz.at

Abstract

Accurate 3D object detection is a critical component of autonomous driving, enabling vehicles to perceive their surroundings with precision and make informed decisions. LiDAR sensors, widely used for their ability to provide detailed 3D measurements, are key to achieving this capability. However, variations between training and inference data can cause significant performance drops when object detection models are employed in different sensor settings. One critical factor is beam density, as inference on sparse, cost-effective LiDAR sensors is often preferred in real-world applications. Despite previous work addressing the beam-density-induced domain gap, substantial knowledge gaps remain, particularly concerning dense 128-beam sensors in cross-domain scenarios.

To gain better understanding of the impact of beam density on domain gaps, we conduct a comprehensive investigation that includes an evaluation of different object detection architectures. Our architecture evaluation reveals that combining voxel- and point-based approaches yields superior cross-domain performance by leveraging the strengths of both representations. Building on these findings, we analyze beam-density-induced domain gaps and argue that these domain gaps must be evaluated in conjunction with other domain shifts. Contrary to conventional beliefs, our experiments reveal that detectors benefit from training on denser data and exhibit robustness to beam density variations during inference.

1. Introduction

Autonomous driving has been receiving increasing attention in recent years, as it has the potential to increase road safety, traffic efficiency, and reduce emissions. To enable the decision-making capabilities of Advanced Driver Assistance Systems (ADAS) or Automated Driving (AD) technologies, understanding the vehicle’s immediate environment is crucial. Light Detection And Ranging (LiDAR) technology stands out as a particularly effective solution for this task through its ability to directly measure three-dimensional distances with high accuracy [27].

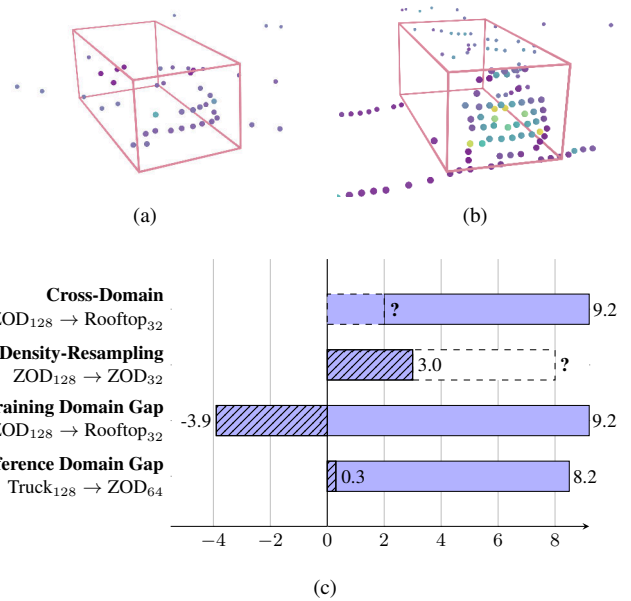


Figure 1. (a) Low-density and (b) high-density scan of vehicles at a similar distance. (c) Overall and beam-density-induced domain gap (in % for IOU=0.4) measured by different methods. The Cross-Domain and Density-Resampling methods fail to assess either the beam-density-induced or overall domain gap, while the Training and Inference Domain Gaps provide a complete picture.

LiDAR-based 3D object detection models have demonstrated impressive performance on established benchmarks [2, 5, 12, 25, 37]. However, their performance often drops significantly when applied across different datasets due to inherent differences between the source domain and the target domain. Typical examples are varying sensor configurations between source and target domain or adverse weather conditions covered by the target but not the source data. When these differences are substantial, the detection model struggles to generalize to the new domain, introducing a performance gap known as the *domain gap*. This challenge is particularly critical in real-world applications, where a domain gap is almost inevitable due to the variability between the training dataset and the diverse conditions encountered in deployment.

The common way to mitigate domain gaps is the appli-

cation of domain adaptation methods. Such methods are oftentimes tailored towards a specific domain difference, such as different LiDAR resolutions [15, 20, 40] or varying object size distributions [23, 39, 44]. Thus, to successfully apply domain adaptation methods, the most significant domain shifts have to be identified first. A structured domain shift taxonomy is useful in this context, as it helps to categorize and systematically understand the specific shifts between domains, enabling the selection or design of a targeted adaptation technique.

Despite taxonomies of related work [6, 8, 24, 47], no study includes all the domain shifts between the datasets under investigation (described in Sec. 3.1). Thus, we introduce a domain shift taxonomy in Tab. 2. Keeping the goal of domain adaptation in mind, we distinguish between domain shifts that can effectively be addressed by domain adaptation methods and those that persist despite the application of domain adaptation. We call the former *non-persistent* and the later *persistent* domain shifts.

Motivated by the observation that the domain gap varies significantly when employing different detectors, we conduct an object detector architecture evaluation. While prior studies do not give particular attention to a thoughtful selection of an object detection model [31], we aim to identify detectors that are inherently robust against domain changes. By minimizing the initial domain gap, the reliance on domain adaptation is minimized, ensuring that the domain adaptation efforts focus on the most challenging aspects of domain gap. We find that (1) voxel-based detectors robustly detect objects, but have difficulties at precisely localizing them and (2) point-based detectors excel at localizing objects in cross-domain settings. Our experiments suggest that optimal cross-domain detection performance is achieved by combining voxel- and point-based approaches in a two-staged detector.

A particularly important domain shift stems from the number of LiDAR beams (see Fig. 1). High-density LiDAR sensors produce detailed point clouds with a high number of points, easing the accurate estimation of object sizes and positions. Low-density LiDAR sensors, which are often more affordable and more commonly used in large-scale deployments, capture fewer points, leading to sparser point clouds and less reliable detection results. This difference in beam density creates a domain shift when models trained on high-density LiDAR data are applied to low-density data and vice versa. To analyze the domain gap caused by varying beam densities, related studies utilize one of two approaches. The first approach [15] involves multiple datasets employing LiDAR sensors with varying beam densities which are subsequently compared. The second approach [8, 31, 40] is based on downsampling a dense dataset to create sparser twin-dataset with varying beam density which are subsequently compared.

This paper highlights the shortcomings of the existing methods. First, comparing the domain gap between two datasets does not guarantee that the observed domain

gap actually stems from varying beam density or is caused by other domain shifts occurring between the investigated datasets. The second approach leads to ambiguous results because it analyzes the effect of beam density in isolation of other domain shifts. In real-world applications, the beam-density-induced domain gap is always accompanied by other effects influencing the domain gap. We show that the beam-density-caused domain gap has to be assessed *in conjunction* with other domain shifts to accurately evaluate its impact in real-world applications. Our experiments suggest that (1) in contrast to the results of related studies [8, 10, 31], it is more beneficial to train object detectors on dense data, independent of the density of the target data and (2) concerning the inference domain gap, detectors are robust against a change of up to 64 beams (see Fig. 1c). Our contributions can be summarized as follows:

- We introduce a domain shift taxonomy based on macro-, sensor-, and object-level domain shifts and distinguish between persistent and non-persistent domain shifts.
- We conduct a detector architecture evaluation where we compare different detectors by their inherent domain adaptation abilities.
- We investigate the domain gap induced by varying beam densities including 128-beam sensors on real-world datasets with consideration of other domain shifts.

2. Related Work

Object Detection: In LiDAR-based 3D object detection, architectural choices heavily influence detection performance. Voxel-based methods [43, 54], discretize LiDAR points into 3D grids, allowing for efficient feature extraction through sparse convolutions. Pillar-based methods [9, 18, 19, 33] convert the point cloud into a 2D BEV-image, sacrificing height information for computational efficiency. Operating directly on the raw points, point-based approaches [28, 30, 52] retain spatial details without quantization. Recent transformer-based models [38, 53, 55], provide an alternative to CNN-based models [3, 49, 50], capturing interactions across larger spatial regions. Detection heads in object detection are either anchor-based [21], relying on predefined anchor sizes, or anchor-free [46], which directly predict object centers to generate bounding boxes. Two-staged detectors [34, 36, 36] split the detection into a proposal and refinement stage, often improving accuracy over single-stage detectors but at a higher computational cost.

Concurrent to our work, Eskandar *et al.* [8] empirically test the impact of fundamental architecture choices. However, they chose different detectors to represent each architectural choice. While Eskandar *et al.* choose PointRCNN [34], VoTr [26] and PV-RCNN [35], we select the faster or better performing object detectors IA-SSD [52], DSVT [38] and PV-RCNN++ [36] for the point-based, Transformer-based and two-staged architectures.

Domain Gap Analysis: Recent works have extensively studied how specific domain shifts contribute to the over-

all domain gap. Wang *et al.* [39] analyzed the impact of geographical variations, concluding that differences in object size distribution can significantly affect detection performance. Another well-studied factor is weather [7, 13]: while LiDAR sensors are less susceptible to adverse weather than cameras, conditions such as snow [17], rain [42], or fog [16] still impair object detection. Concerning sensor-level domain shifts, Hu *et al.* [14] and Fang *et al.* [10] investigate the impact of different LiDAR mounting positions. There are also some recent works investigating the effect of varying beam densities [10, 31]. Richter *et al.* [31] perform a real-world study comparing a 32-beam and 64-beam LiDAR sensors utilizing a specially designed dataset, isolating the beam-density-induced domain gap. However, they do not analyze beam density in conjunction with other domain shifts such as geographic location or object size. Fang *et al.* [10] perform a systematic study regarding beam density on a simulated dataset. However, they did not test the transferability of their findings to real-world datasets.

Domain Adaptation: Domain adaptation methods aim to improve object detection performance across different datasets, addressing the challenges introduced by domain shifts. Broadly, domain adaptation approaches fall into one of three categories: domain alignment, feature alignment, and self-training.

The domain alignment methods SN [39], OT [39] and SAILOR [23] excel at handling object size discrepancies by rescaling ground truth bounding boxes during training or inference. For beam density shifts, methods like DTS [15], PDDA [20] and LiDAR-CS [10] employ re-sampling methods to align point cloud densities. ReSimAD [48] aligns more complex LiDAR sensor characteristics by reconstructing target scenes and rendering source-like point clouds. Feature alignment methods [22, 40, 41, 51], another approach, perform domain adaptation by alignment in feature space instead of aligning the point clouds directly. In self-training [4, 11, 29, 32, 44, 45], iterative refinement of pseudo-labels is used to gradually adapt the detector to the target domain.

While these domain adaptation methods effectively reduce the occurring domain gaps, they pay little attention to the underlying object detector. We show that a thoughtful selection of the object detector architecture can already close a portion of the domain gap, which reduces the reliance on domain adaptation methods and shifts the focus to more complex domain shifts which cannot be mitigated through architecture alone.

3. Preliminary Analysis

Our preliminary analysis lays the groundwork for this study by addressing three aspects. First, we introduce the non-public datasets involved in this study and detail their unique properties. Second, we establish a domain shift taxonomy, allowing us to systematically assess domain differences. Third, we conduct a detector architecture evaluation to identify models that are inherently ro-

	Truck	Rooftop	ZOD
Locations	Germany	Germany	15 European Countries
Ann. frames	40k	7.5k	100k
Sequences	2036	251	43468
Top LiDAR	OS2 (128-beam)	VLP 32c (32-beam)	VLS 128 (128-beam)
Mounting height	3.41m	1.78m	2.01m
Side LiDARs	64-beam	16-beam	16-beam
Front LiDAR	32-beam	-	-
Avg. pts per frame	178.4k	71.5k	254k
Points per beam	2048	1800	3270
Horizontal res.	0.18°	0.2°	0.11°
License	private	private	CC BY-SA

Table 1. Dataset overview.

bust to domain shifts, providing a foundation for effective domain adaptation.

3.1. Dataset Introduction

In this paper we leverage three datasets (see Tab. 1 and Fig. 2) for training and evaluation. The Rooftop and Truck datasets are private while the remaining Zenseact Open Dataset (ZOD) [1] is open-source. All datasets are specifically designed for autonomous driving applications and feature frame-wise LiDAR data. Concerning the dataset size, the Rooftop dataset is the smallest, with about 7.5k annotated frames, while the Truck and Zenseact Open datasets are substantially larger with 40k and 100k annotated frames. While the Rooftop and Truck datasets were both recorded in Germany, the ZOD contains data from 15 different European countries. Another substantial difference concerns the organization of frames. The Rooftop and Truck datasets are structured in sequences of 20 or 30 frames per sequence, while the ZOD consists of single frames, where, on average, only two frames belong to the same sequence. Regarding the sensor setup, the main LiDAR also differs between each dataset. The Zenseact and Truck datasets employ a dense 128-beam LiDAR, each from a different manufacturer, while the Rooftop dataset employs a sparse 32-beam main LiDAR. We conduct a detailed analysis of the differences between the datasets in the subsequent Sec. 3.2.

Some inherent differences between the datasets can be eliminated by dataset alignment. The size difference can be aligned rather easily by randomly subsampling of the larger datasets to match the size of the smallest dataset. We identified three major dataset alignment measures to address the frame content: coordinate, range and label-space alignment.

Coordinate Alignment: Datasets often differ in coordinate systems, leading to potential mismatches between LiDAR points and object labels. To address this, we align all data points and labels with the commonly used sensor coordinate system with a forward-pointing x-axis and

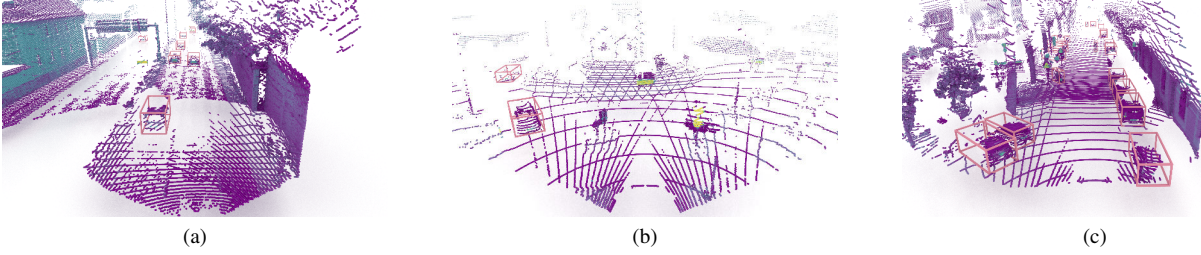


Figure 2. Comparison of the (a) Truck, (b) Rooftop and (c) Zenseact datasets. Differences in LiDAR beam density are clearly visible. Ground-truth objects are marked by red bounding boxes.

	Persistent	Non-persistent
Macro-level	Collection Area Type	Object Size Statistics
	Geographical Location	Weather Conditions
	Frame Selection	
Sensor-level	Sensor Setup	Beam Density
	Intensity Value	Horizontal Resolution
	Rate of Rotation	Field of View
	Alignment Error	
Object-level	Labeling Quality	Label Space Definition
		Labeling Zone
		Object Definition

Table 2. Domain shift taxonomy. We differentiate between persistent and non-persistent to highlight domain shifts that can effectively be addressed by either readily available domain adaptation methods or dataset alignment measures.

an upward-pointing z-axis. Since sensors are mounted at different heights, we standardize the origin by aligning it with the ground plane.

Range Alignment: Standardizing the Field of View (FOV) across datasets allows the detector to learn consistent object regions. We define a forward detection range of 123.2 meters to support high-speed safety applications and limit the horizontal FOV to 120° to match the ZOD’s labeled region. We mark objects truncated by the cropped FOV as “ignore” during training and evaluation, which prevents the generation of a loss from these objects.

Label-Space Alignment: Inherent label-space differences of the ground-truth annotations between datasets necessitate a mapping to standardize object classes. We categorize objects into four primary classes: *Vehicle*, *Truck*, *Single-track*, and *Pedestrian*. Single-track vehicles are composed of bicycle and motorcycles. Larger vehicles such as vans, trucks and trailers fall under the *Truck* class. To handle varying labeling conventions concerning single-tracked vehicles and their riders, we merge their bounding boxes encompassing both as a single object. A detailed mapping of the label-spaces between the three datasets can be found in the Supplementary.

3.2. Domain Shift Taxonomy

We propose a domain shift taxonomy which allows for a detailed and systematic investigation of possibly occurring domain shifts between aligned datasets. We distin-

guish between three main categories. Sensor-level domain shifts are directly caused by the mode of collection, while Object-level domain shifts concern the object definition and labeling. The remaining macro-level domain shifts are mainly caused by differences in dataset content. Keeping the final goal of domain adaptation in mind, we additionally differentiate between domain shifts that can effectively be reduced by domain adaptation methods, the *non-persistent domain shifts*, and those that persist despite domain adaptation methods, which we refer to as *persistent domain shifts*.

Concerning the *persistent domain shifts*, we notice a few macro-level differences. While the ZOD features a geographically diverse set of recording locations, the Rooftop and Truck datasets were exclusively recorded in Germany. Also, the types of areas differ between datasets: the ZOD features substantially more *City* frames compared to the remaining two datasets. Finally, we find differences that likely originate from the frame selection process for each dataset. We notice that there is a significantly lower number of overall objects in the Rooftop dataset compared to the Truck and Zenseact datasets. Especially the *Pedestrian* and *Cyclist* classes are significantly underrepresented such that the missing diversity of classes would dominate the domain gap. Thus, we resort to mainly perform dataset-wise comparisons between the *Vehicle* classes.

There are also significant differences on a sensor-level. The Truck dataset has a unique sensor setup as the sensors are mounted considerably higher compared to the other two datasets. The high mounting position has the consequence of a large blind spot right in front of the ego vehicle. The installation of an additional forward-facing LiDAR addresses this issue, resulting in a four-sensor setup.

In terms of object-level differences, we find disparities between the datasets caused by deficient labeling. More specifically, we notice missing ground truth labels for the Rooftop dataset, especially for distant objects that are hit by less LiDAR points. The implications are a noisy supervision signal for training and a distorted evaluation result as predominantly hard-to-detect objects are missing. The ZOD suffers from a similar problem, but hereby, the missing labels are caused by the labeling procedure. ZOD’s labeling is based on the camera images. Slight height differences between the camera and LiDAR sensors cause objects to be occluded for the camera while visible for the

Detector	Backbone Architecture		Detection Head	Stages
SECOND [43]	Voxel	CNN	Anchor	Single
PointPillar [18]	Pillar	CNN	Anchor	Single
IA-SSD [52]	Point	CNN	Point	Single
CenterPoint [46]	Voxel	CNN	Center	Single
PVRCNN++ [36]	Point-Voxel	CNN	Center/Point	Two
DSVT [38]	Pillar	Trans-former	Center	Single

Table 3. List of 3D object detection methods and their architectural properties.

LiDAR, resulting in missing labels.

We also identify many *non-persistent domain shifts*. In contrast to the previous class of domain shifts, the non-persistent ones can effectively be reduced or even eliminated by domain adaptation methods. Most prominently, the datasets employ LiDAR sensors with a differing number of beams as well as varying beam patterns. Furthermore, the ZOD is more diverse in terms of captured weather conditions as it also features adverse weather conditions such as fog or snow. We also notice differences in terms of object sizes. As the ZOD contains frames recorded in multiple different countries, the intra-dataset object size variability is higher.

Throughout our analysis, we find numerous domain shifts between the datasets. Most of the identified shifts cannot be isolated, making it infeasible to estimate the impact of individual domain shifts on the overall domain gap by a simple comparison between datasets. We provide detailed statistics and domain shift examples in the Supplementary Material.

3.3. Detector Architecture Evaluation

We identify six key differences among commonly used object detection architectures and select one object detector representative of each difference. This approach allows us to assess the impact of each architectural choice. An overview of the selected object detectors is given in Tab. 3. In terms of data representation, we choose SECOND [43] to represent voxel-based architectures, PointPillars [18] for the pillar-based representation, and IA-SSD [52] to represent the class of point-based object detectors. Furthermore, we select CenterPoint [46] to assess the effect of center-based detection heads. To reason about the effectiveness of two-staged methods, we employ the point-voxel-based detector PV-RCNN++ [36]. This detector uses a SECOND-like first-stage to extract bounding box proposals and a point-feature-based second-stage to refine the proposals for the final bounding box estimation. Lastly, we test the impact of different feature extractor architectures. As Transformer-based architectures have recently established themselves in the field of 3D object detection [38], we test their performance in comparison

to the well-established sparse-convolution-based architectures.

4. Approach

To evaluate the impact of the beam density on the cross-domain performance, we first select a detector architecture that demonstrates robustness across domains. In our initial experiments, we simply evaluate the trained detectors across domains and group the detection results according to the domain shifts of interest. Results are reported in both high- and low-IOU settings to differentiate between localization and detection errors. For our analysis, we primarily focus on detection errors, which are assessed using low-IOU experiments, as localization errors can usually be mitigated through domain adaptation methods targeting object sizes [23, 39]. This cross-domain comparison operates under the assumption that the datasets are sufficiently similar to enable meaningful conclusions. However, as elaborated in Sec. 3.2, this assumption rarely holds due to persistent domain gaps arising from differences in sensor setups, environmental conditions, and other factors.

In our second set of experiments, we analyze the impact of varying beam densities by isolating it from other domain shifts. Following the approach of [8, 40], we generate beam-wise downsampled versions of one dataset. A detector is trained on each version and subsequently evaluated on the other versions of the same dataset. This approach allows us to focus on the impact of beam density, independent of other domain-specific properties. However, in real-world applications, a varying beam density is usually just one of many domain shifts occurring at test time. In such cases, other kinds of domain shifts may completely dominate the domain gap, rendering the effect of beam density negligible. On the contrary, it could also be the case that due to the cross-domain application, other more reliable features are missing, resulting in an increased domain gap caused by varying beam density. Focusing on a single domain shift in isolation fails to capture these complex interactions. Thus, more sophisticated experiments with the goal of capturing the domain gap by a certain domain shift *in conjunction* with other domain shifts is necessary.

To address the limitations of isolated domain shift analysis, we propose an experimental setup designed to account for interactions between beam density and other domain shifts. As in prior experiments, we utilize sparsified versions of datasets to analyze the impact of beam density, but this time in conjunction with other datasets. Our setup divides the domain gap into two components: the *training domain gap*, caused by differences in beam density during training, and the *inference domain gap*, caused by variations during evaluation, as described by Richter *et al.* [31]. To measure the training domain gap, we downsample the training dataset to create multiple versions, each representing a specific beam density level. By matching or mismatching the beam density with the evaluation dataset, we

isolate the effects of beam density variation during training. Similarly, the inference domain gap can be attained by varying the beam density of the evaluation datasets while keeping the training datasets unchanged. These controlled experiments allow us to isolate the specific effects of beam density in a cross-domain setting. We note that in real-world applications, it is typically infeasible to disentangle the training and inference domain gaps, underscoring the relevance of these controlled experiments.

5. Experiments

To demonstrate the effectiveness of our assessment approach, we conduct experiments on the two private datasets Rooftop and Truck and the public Zenseact Open Dataset [1]. We first present our results for the object detector architecture evaluation, based on which we then assess the cross-domain and density-resampling domain gaps. Finally, we compare our training and inference domain gaps to the previously determined domain gaps. Detailed experiment results and additional information about implementation and model training can be found in the Supplementary Material.

5.1. Evaluation Metrics

We use the Intersection-over-Union (IOU)-based metric *3D average precision* $AP_{S \rightarrow T}$ to assess the detection performance of an object detection model trained on the source domain \mathbb{D}^S when evaluated on the target domain \mathbb{D}^T . By lowering the IOU threshold, we can additionally disentangle the *detection error* from the *localization error*. Thereby, localization errors are caused by objects that are detected but not localized accurately enough to be considered true positives, whereas detection errors represent entirely missed or wrongly classified objects. Wang *et al.* [39] demonstrated that the 3D average precision significantly increases when the IOU threshold is reduced from the commonly used threshold of 0.7 (70%) to approximately 0.4 (40%). At this threshold, the domain gap primarily reflects detection errors, which are of greater practical significance than localization errors. Localization errors can often be mitigated using domain adaptation methods such as ROS [44], SN, or OT [39]. Therefore, in our evaluation, we primarily focus on a reduced IOU threshold of 0.4 to better understand detection errors.

While the cross-domain performance is well suited for comparing different detectors, it does not adequately capture the generalization ability of a certain detector across domains, as it is influenced by the inherent difficulty of the target dataset. To address this limitation, we employ the *domain gap* metric [44], which relates cross-domain performance to the detector’s maximum achievable performance on the target domain ($AP_{T \rightarrow T}$). This relative metric provides a detached view of domain generalization ability and allows for meaningful comparisons of detectors evaluated on target datasets with varying difficulty levels. The domain gap DG , expressed as a percentage of the maximum achievable performance, is defined as

Detector	mAP \uparrow IOU=0.7	mAP \uparrow IOU=0.4	DG in % \downarrow IOU=0.7	DG in % \downarrow IOU=0.4
SECOND	30.6	70.4	47.6	16.0
PointPillars	23.0	63.9	56.4	21.5
IA-SSD	34.7	66.1	41.5	18.3
CenterPoint	28.6	68.1	49.7	18.0
PV-RCNN++	37.2	71.2	42.8	16.2
DSVT	33.4	68.4	47.7	20.1

Table 4. Detector comparison results overview. We calculate the cross-domain performance by averaging over all cross-domain results. We report the average domain gap and the cross-domain performance using the AP metric at the IOU thresholds of 0.7 and 0.4 for the *Vehicle* class.

$$DG = \frac{AP_{T \rightarrow T} - AP_{S \rightarrow T}}{AP_{T \rightarrow T}} \cdot 100 \quad (1)$$

5.2. Detector Architecture Evaluation

With the goal of finding a detector that is robust against domain changes, we examine the impact of each architectural choice on the overall performance in the cross-domain setting (see Tab. 4). The first architectural comparison concerns the voxel and pillar discretization methods. We find that the voxel-based detector SECOND [43] outperforms the pillar-based PointPillars [18] by a substantial margin. While PointPillars is the worst-performing detector across all metrics, SECOND exhibits surprisingly good performance in the low-IOU settings. This indicates that SECOND is good at detecting objects but fails to precisely locate them in 3D space. This discrepancy stems from the quantization process. During voxelization, the exact geometric structure is lost, hampering the precise localization of objects. In terms of different data representations, we also test the point-based detector IA-SSD [52]. This detector shows a very strong performance in the high-IOU setting, indicating that it is also good at predicting the 3D location of objects. This can be attributed to the detector’s direct access to the point data.

Subsequently, we test the effects of different detection heads. More specifically, we compare anchor-heads, as employed in SECOND or PointPillars, with center-heads, as introduced in CenterPoint [46]. Contrary to expectations, center-heads result in degraded performance compared to anchor-heads.

Next, we apply PV-RCNN++ [36] to test the impact of an additional second stage. This detector outperforms all others in terms of cross-domain performance while staying competitive in terms of domain gap. Similar to IA-SSD, the second stage of PV-RCNN++ benefits from direct access to raw point data, which likely enhances its performance. We can conclude that, for our experiments, the addition of a second stage significantly benefits the object detectors regarding generalization abilities.

Lastly, we examine the effect of Transformer-based backbones. While DSVT [38] achieves excellent in-domain results, its cross-domain performance and domain

Source→Target		avg. DG in % IOU=0.7 ↓	avg. DG in % IOU=0.4 ↓
Beam Density	Dense→Dense	23.9	14.4
	Dense→Sparse	36.4	11.1
	Sparse→Dense	67.4	23.0

Table 5. Domain gap in percent for the object detector PV-RCNN++. The different cross-domain settings are grouped and averaged by beam density. We report the average domain gap calculated with the AP metric at the IOU thresholds of 0.7 and 0.4 for the *Vehicle* class.

gap metrics are only moderate. Our experiments suggest that Transformer-based backbones do not benefit object detectors in terms of domain generalization.

These findings highlight the critical role of detector architecture in achieving robust domain generalization. Comparing the best and worst-performing detectors, we observe a performance difference of 61.7% in the high-IOU setting and 11.4% in the low-IOU settings. We further find that purely voxel-based detectors excel at detecting objects and the addition of point information drastically improves the localization error. The object detection architecture evaluation conducted by Eskandar *et al.* [8] yields similar conclusion concerning the effect of point information. However, our experiments do not support their finding that Transformer-based backbones improve cross-domain generalization.

5.3. Cross-domain Results

Beginning our examination of the domain gap induced by beam density, we conduct a simple cross-domain evaluation. As shown in Tab. 5, the sparsely-trained detector (trained on Rooftop) applied on denser datasets (Sparse→Dense) exhibits approximately twice the domain gap compared to applying a densely-trained detector (trained on Truck or ZOD) on a sparse dataset (Dense→Sparse). This trend persists when isolating the detection error by evaluating with reduced IOU threshold. In terms of domain generalization, we could conclude from this initial analysis that for these particular datasets, it is beneficial to train a detector on the dense datasets as they generalize towards sparse and dense datasets. As the effect of beam density is just one of many factors contributing to this observed domain gap, further analysis is required to make stronger statements.

5.4. Density-resampling Results

We continue by isolating the beam-density-induced domain gap in our second set of experiments. To keep track of the resampled density, we call the original dense dataset ZOD₁₂₈ and the sparser variants ZOD₆₄ and ZOD₃₂, where the index represents the number of beams. As shown in Tab. 6, the density-resampling analysis contrasts with the findings from the cross-domain analysis. In the high-IOU setting, the sparse-to-dense cases (top-right of the results matrix) give better results than their dense-

Source	Target		
	ZOD ₃₂	ZOD ₆₄	ZOD ₁₂₈
	DG in % ↓ IOU=0.7	DG in % ↓ IOU=0.7	DG in % ↓ IOU=0.7
ZOD ₃₂	-	1.8	5.1
ZOD ₆₄	4.3	-	1.9
ZOD ₁₂₈	9.4	0.9	-
	DG in % ↓ IOU=0.4	DG in % ↓ IOU=0.4	DG in % ↓ IOU=0.4
	ZOD ₃₂	ZOD ₆₄	ZOD ₁₂₈
	DG in % ↓ IOU=0.4	DG in % ↓ IOU=0.4	DG in % ↓ IOU=0.4
ZOD ₃₂	-	0.6	2.3
ZOD ₆₄	2.7	-	1.6
ZOD ₁₂₈	3.0	-0.9	-

Table 6. Density-caused domain gap for the density-resampling setting. We report the domain gap calculated with the AP metric at the IOU thresholds of 0.7 (top) and 0.4 (bottom) for the *Vehicle* class.

to-sparse counterparts (bottom-left of the results matrix). More broadly, we notice that the performance differences between all datasets are comparably small. This indicates that the detectors generalize very well towards the same dataset when solely varying the sampling. However, in real-world applications, variations in sampling is usually accompanied by other kinds of domain shift. In the following experiments we thus investigate the beam-density-induced domain shift in the presence of other kinds of domain shifts by measuring the training and inference domain gaps.

5.5. Training and Inference Domain Gap Results

We first examine the training domain gap in Tab. 8. Compared to the density-resampling setting (recall Tab. 6), the overall domain gap level is significantly higher, as many more factors contribute besides the beam density. In the high-IOU experiments, trends are consistent with the density-resampling case: larger differences in beam density lead to larger domain gaps. However, when isolating the detection error in the low-IOU setting, a different trend emerges. Densely-trained detectors show overall better performance for the dense *and* sparse target datasets, indicating that they are able to detect more objects than their sparsely-trained counterparts in the cross-domain case. This setup also allows us to quantify the impact of beam density on the overall domain gap. When evaluating on the Rooftop dataset, training on a denser 128-beam dataset *reduces* the domain gap from 13.1% to 9.2%, reducing the domain shift by almost one-third (shown in Fig. 1c).

Next, we analyze the inference domain gap in Tab. 7. The high-IOU results (top-left of the table) show greater domain gap variability between the Rooftop and Truck datasets than within each dataset across beam densities. This observation supports our earlier assumption about persistent dataset-caused domain gaps in cross-domain evaluation settings (recall Sec. 4). Overall, the results in-

Source	Target						
	ZOD₃₂	ZOD₆₄	ZOD₁₂₈	ZOD₃₂	ZOD₆₄	ZOD₁₂₈	
	DG in % ↓	DG in % ↓	DG in % ↓	DG in % ↓	DG in % ↓	DG in % ↓	
	IOU=0.7	IOU=0.7	IOU=0.7	IOU=0.4	IOU=0.4	IOU=0.4	
Rooftop₃₂	55.0	58.0	67.4	13.9	11.8	13.4	
Truck₁₂₈	28.6	19.0	16.3	16.9	8.5	8.2	
	AP ↑	AP ↑	AP ↑	AP ↑	AP ↑	AP ↑	
	IOU=0.7	IOU=0.7	IOU=0.7	IOU=0.4	IOU=0.4	IOU=0.4	
	Rooftop₃₂	25.1	26.7	22.4	60.5	69.0	73.0
	Truck₁₂₈	39.7	51.4	57.6	58.4	71.6	77.4

Table 7. Inference domain gap caused by varying beam densities in a cross-domain setting. We report the domain gap (top) and the cross-domain performance (bottom) using the AP metric at the IOU thresholds of 0.7 (left) and 0.4 (right) for the *Vehicle* class.

Source	Target			
	Rooftop₃₂	Truck₁₂₈	Rooftop₃₂	Truck₁₂₈
	DG in % ↓ IOU=0.7	DG in % ↓ IOU=0.7	DG in % ↓ IOU=0.4	DG in % ↓ IOU=0.4
ZOD₃₂	38.4	41.5	13.1	28.0
ZOD₆₄	41.2	37.2	11.1	23.7
ZOD₁₂₈	45.8	32.5	9.2	20.5

Table 8. Training domain gap caused by varying beam densities in a cross-domain setting. We report the domain gap calculated with the AP metric at the IOU thresholds of 0.7 (left) and 0.4 (right) for the *Vehicle* class.

dicates that the detector is relatively robust to beam density, provided the number of beams does not change drastically. In the high-IOU experiments, the domain gap increases by only 3% or less when doubling or halving beam density.

The previous analyses were exclusively done through the lens of the *domain gap* metric. For the inference domain gap, it is also interesting to examine the performance values themselves. Especially for the low-IOU setting, we can see in Tab. 7 that the performance (measured in AP) increases steadily with an increasing number of beams, despite the domain gap staying similar. This indicates that the observed performance gain is caused by easing the detection problem in contrast to better generalizability of the detectors. As the density increases, more LiDAR rays hit objects which makes it easier for the object to be detected.

In summary, the results provide a comprehensive view of the domain gap caused by varying beam densities. When isolating the effect of varying beam densities (see Tab. 6), the domain gap appears minor, favoring sparsely-trained detectors for domain generalization. This aligns with findings from related studies [8, 10, 31]. However, when analyzing the training domain gap in conjunction with other domain shifts, we find that densely-trained detectors exhibit better domain generalization in terms of detecting objects (see Tab. 8). Regarding inference domain gaps (see Tab. 7), results show that detectors gen-

eralize well as long as beam density changes are modest. Nonetheless, denser sampling reduces detection difficulty, leading to better performance irrespective of the detector’s generalizability.

6. Conclusion

This study presented an investigation of the impact of beam density on LiDAR object detection performance in cross-domain scenarios during which we also explored optimal object detector architectures to address domain variability effectively. Our object detector architecture evaluation revealed that combining voxel- and point-based approaches delivers superior cross-domain performance by leveraging the complementary strengths of these representations. While Transformer-based backbones demonstrated strong performance in in-domain tasks, their cross-domain benefits were limited under the conditions tested. Our findings emphasize the importance of selecting a robust detector architecture as a prior step to domain adaptation.

We further investigated the impact of beam density on LiDAR object detection performance in cross-domain scenarios, offering insights into both training and inference domain gaps. We found that detectors trained on dense datasets generalize better across domains, particularly for detecting objects, where detection error (rather than localization error) is the primary concern. During inference, detectors showed robustness against moderate beam density changes, with denser configurations improving performance by reducing the difficulty of the detection task rather than enhancing generalizability.

A key insight from this study is that domain gaps, including those caused by beam density, should not be analyzed in isolation. Instead, we advocate for a holistic approach to domain adaptation, beginning with the selection of a detector intrinsically robust to domain changes. This minimizes the initial domain gap and allows adaptation efforts to focus on more complex types of domain shifts.

7. Acknowledgements

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

References

- [1] Mina Alibeigi, William Ljungbergh, Adam Tonderski, Georg Hess, Adam Lilja, Carl Lindstrom, Daria Motorniuk, Junsheng Fu, Jenny Widahl, and Christoffer Petersson. Zenseact Open Dataset: A large-scale and diverse multimodal dataset for autonomous driving. In *Proc. CVPR*, 2023. 3, 6
- [2] 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 *Proc. CVPR*, 2020. 1
- [3] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. LargeKernel3D: Scaling up Kernels in 3D Sparse CNNs. In *Proc. CVPR*, 2023. 2
- [4] Zhuoxiao Chen, Yadan Luo, Zheng Wang, Mahsa Baktashmotlagh, and Zi Huang. Revisiting Domain-Adaptive 3D Object Detection by Reliable, Diverse and Class-balanced Pseudo-Labeling. In *Proc. ICCV*, 2023. 3
- [5] Carlos A. Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, Wei-Lun Chao, Bharath Hariharan, Kilian Q. Weinberger, and Mark Campbell. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In *Proc. CVPR*, 2022. 1
- [6] Yinpeng Dong, Caixin Kang, Jinlai Zhang, Zijian Zhu, Yikai Wang, Xiao Yang, Hang Su, Xingxing Wei, and Jun Zhu. Benchmarking robustness of 3d object detection to common corruptions in autonomous driving. In *Proc. CVPR*, 2023. 2
- [7] Mariella Dreissig, Dominik Scheuble, Florian Piewak, and Joschka Boedecker. Survey on LiDAR Perception in Adverse Weather Conditions. In *IEEE Intelligent Vehicles Symposium*, 2023. 3
- [8] George Eskandar, Chongzhe Zhang, Abhishek Kaushik, Karim Guirguis, Mohamed Sayed, and Bin Yang. An Empirical Study of the Generalization Ability of Lidar 3D Object Detectors to Unseen Domains. In *Proc. CVPR*, 2024. 2, 5, 7, 8
- [9] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing Single Stride 3D Object Detector with Sparse Transformer. In *Proc. CVPR*, 2022. 2
- [10] Jin Fang, Dingfu Zhou, Jingjing Zhao, Chenming Wu, Chulin Tang, Cheng-Zhong Xu, and Liangjun Zhang. LiDAR-CS Dataset: LiDAR Point Cloud Dataset with Cross-Sensors for 3D Object Detection. In *Proc. ICRA*, 2024. 2, 3, 8
- [11] Christian Fruhwirth-Reisinger, Michael Opitz, Horst Possegger, and Horst Bischof. FAST3D: Flow-Aware Self-Training for 3D Object Detectors. In *Proc. BMVC*, 2021. 3
- [12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. CVPR*, 2012. 1
- [13] Deepti Hegde, Velat Kilic, Vishwanath Sindagi, A Branton Cooper, Mark Foster, and Vishal M Patel. Source-free unsupervised domain adaptation for 3d object detection in adverse weather. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 3
- [14] Hanjiang Hu, Zuxin Liu, Sharad Chitlangia, Akhil Agnihotri, and Ding Zhao. Investigating the Impact of Multi-LiDAR Placement on Object Detection for Autonomous Driving. In *Proc. CVPR*, 2022. 3
- [15] Qianjiang Hu, Daizong Liu, and Wei Hu. Density-Insensitive Unsupervised Domain Adaption on 3D Object Detection. In *Proc. CVPR*, 2023. 2, 3
- [16] Jiyeon Kim, Bum-jin Park, and Jisoo Kim. Empirical Analysis of Autonomous Vehicle’s LiDAR Detection Performance Degradation for Actual Road Driving in Rain and Fog. *Sensors*, 2023. 3
- [17] Akhil Kurup and Jeremy Bos. Dsor: A scalable statistical filter for removing falling snow from lidar point clouds in severe winter weather, 2021. 3
- [18] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *Proc. CVPR*, 2019. 2, 5, 6
- [19] Jinyu Li, Chenxu Luo, and Xiaodong Yang. PillarNeXt: Rethinking Network Designs for 3D Object Detection in LiDAR Point Clouds. In *Proc. CVPR*, 2023. 2
- [20] Shuangzhi Li, Lei Ma, and Xingyu Li. Domain generalization of 3d object detection by density-resampling. In *Proc. ECCV*, 2025. 2, 3
- [21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In *Proc. ECCV*, 2016. 2
- [22] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised Domain Adaptive 3D Detection with Multi-Level Consistency. In *Proc. ICCV*, 2021. 3
- [23] Dusan Malic, Christian Fruhwirth-Reisinger, Horst Possegger, and Horst Bischof. SAILOR: Scaling Anchors via Insights into Latent Object Representation. In *Proc. WACV*, 2023. 2, 3, 5
- [24] Sivabalan Manivasagam, Ioan Andrei Bârsan, Jingkang Wang, Ze Yang, and Raquel Urtasun. Towards zero domain gap: A comprehensive study of realistic LiDAR simulation for autonomy testing. In *Proc. ICCV*, 2023. 2
- [25] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Hanxue Liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu. One million scenes for autonomous driving: ONCE dataset. In *Proc. NeurIPS*, 2021. 1
- [26] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel Transformer for 3D Object Detection. In *Proc. ICCV*, 2021. 2
- [27] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3D Object Detection for Autonomous Driving: A Review and New Outlooks. In *IJCV*, 2022. 1

- [28] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3D Object Detection with Pointformer. In *Proc. CVPR*, 2021. 2
- [29] Xidong Peng, Xinge Zhu, and Yuexin Ma. CL3D: Unsupervised Domain Adaptation for Cross-LiDAR 3D Detection. In *Proc. AAAI*, 2023. 3
- [30] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. CVPR*, 2017. 2
- [31] Jasmine Richter, F. Faion, Di Feng, Paul Benedikt Becker, Piotr Sielecki, and Claudius Glaeser. Understanding the Domain Gap in LiDAR Object Detection Networks. In *Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*, 2022. 2, 3, 5, 8
- [32] Cristiano Saltori, Stephane Lathuiliere, Nicu Sebe, Elisa Ricci, and Fabio Galasso. SF-UDA^{3d}: Source-Free Unsupervised Domain Adaptation for LiDAR-Based 3D Object Detection. In *IEEE 3DV*, 2020. 3
- [33] Guangsheng Shi, Ruifeng Li, and Chao Ma. PillarNet: Real-Time and High-Performance Pillar-based 3D Object Detection. In *Proc. ECCV*, 2022. 2
- [34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *Proc. CVPR*, 2019. 2
- [35] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *Proc. CVPR*, 2020. 2
- [36] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. In *IJCV*, 2022. 2, 5, 6
- [37] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. CVPR*, 2020. 1
- [38] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. DSVT: Dynamic Sparse Voxel Transformer with Rotated Sets. In *Proc. CVPR*, 2023. 2, 5, 6
- [39] Yan Wang, Xiangyu Chen, Yurong You, Li Erran, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in Germany, Test in The USA: Making 3D Object Detectors Generalize. In *Proc. CVPR*, 2020. 2, 3, 5, 6
- [40] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. LiDAR Distillation: Bridging the Beam-Induced Domain Gap for 3D Object Detection. In *Proc. ECCV*, 2022. 2, 3, 5
- [41] Maciej K Wozniak, Mattias Hansson, Marko Thiel, and Patric Jensfelt. Uada3d: Unsupervised adversarial domain adaptation for 3d object detection with sparse lidar and large domain gaps. *IEEE RA-L*, 2024. 3
- [42] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R. Qi, and Dragomir Anguelov. SPG: Unsupervised Domain Adaptation for 3D Object Detection via Semantic Point Generation. In *Proc. ICCV*, 2021. 3
- [43] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10), 2018. 2, 5, 6
- [44] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection. In *Proc. CVPR*, 2021. 2, 3, 6
- [45] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D++: Denoised Self-Training for Unsupervised Domain Adaptation on 3D Object Detection. In *IEEE TPAMI*, 2022. 3
- [46] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Centerpoint: Center-based 3D Object Detection and Tracking. In *Proc. CVPR*, 2021. 2, 5, 6
- [47] Kaicheng Yu, Tang Tao, Hongwei Xie, Zhiwei Lin, Tingting Liang, Bing Wang, Peng Chen, Dayang Hao, Yongtao Wang, and Xiaodan Liang. Benchmarking the robustness of LiDAR-camera fusion for 3d object detection. In *Proc. CVPRW*, 2023. 2
- [48] Bo Zhang, Xinyu Cai, Jiakang Yuan, Donglin Yang, Jianfei Guo, Renqiu Xia, Botian Shi, Min Dou, Tao Chen, Si Liu, Junchi Yan, and Yu Qiao. ReSimAD: Zero-Shot 3D Domain Transfer for Autonomous Driving with Source Reconstruction and Target Simulation. In *Proc. ICLR*, 2024. 3
- [49] Gang Zhang, Junnan Chen, Guohuan Gao, Jianmin Li, and Xiaolin Hu. HEDNet: A Hierarchical Encoder-Decoder Network for 3D Object Detection in Point Clouds. In *Proc. NeurIPS*, 2023. 2
- [50] Gang Zhang, Junnan Chen, Guohuan Gao, Jianmin Li, Si Liu, and Xiaolin Hu. SAFDNet: A Simple and Effective Network for Fully Sparse 3D Object Detection. In *Proc. CVPR*, 2024. 2
- [51] Weichen Zhang, Wen Li, and Dong Xu. SRDAN: Scale-aware and Range-aware Domain Adaptation Network for Cross-dataset 3D Object Detection. In *Proc. CVPR*, 2021. 3
- [52] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not All Points Are Equal: Learning Highly Efficient Point-based Detectors for 3D LiDAR Point Clouds. In *Proc. CVPR*, 2022. 2, 5, 6
- [53] Chao Zhou, Yanan Zhang, Jiaxin Chen, and Di Huang. OcTr: Octree-based Transformer for 3D Object Detection. In *Proc. CVPR*, 2023. 2
- [54] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *Proc. CVPR*, 2018. 2
- [55] Zixiang Zhou, Dongqiangzi Ye, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarformer: A unified transformer-based multi-task network for lidar perception. In *Proc. ICRA*, 2024. 2

An Investigation of Beam Density on LiDAR Object Detection Performance

Supplementary Material

1. Dataset Introduction

In the following, we provide the detailed label map (see Tab. 1) from Chap. 3.1 and elaborate some of the domain shifts mentioned in Chap. 3.2 in more detail. We begin by showcasing the effect of geographically diverse locations in Fig. 1. The observed country-level size bias combined with a dataset-specific size-bias results in different average object sizes between the datasets (see Fig. 2). Fig. 3 shows differences in recording locations. We can see that the ZOD contains significantly more *City* frames compared to the other two datasets. As a consequence, a detector trained on the ZOD is more likely to assign objects that typically associated with *City* frames, such as *Cyclists*, to ambiguous objects than detectors trained on the other datasets. An example of this phenomenon is depicted in Fig. 4, where the detector trained on the ZOD detects a *Cyclist*, while the other detectors correctly detect a *Truck*. The bias introduced by the frame selection procedure can be seen in Fig. 5. The Rooftop dataset contains, on average, less objects per frame than the other two datasets. This difference is especially severe for the classes *Pedestrian* and *Cyclist*. Finally, we give an example for imperfect labeling of the Rooftop dataset in Fig. 6. The camera image shows a black car, which is captured by 8 points in the LiDAR image. However, no bounding box is assigned in the LiDAR frame.

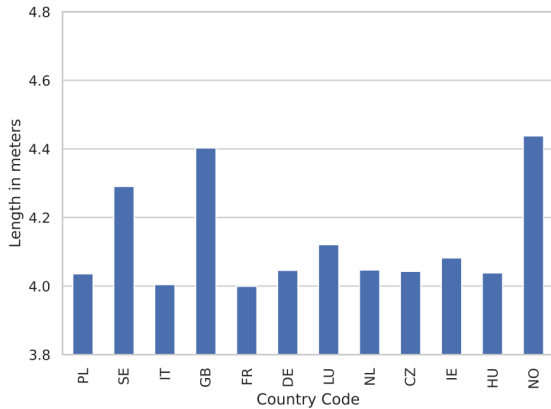


Figure 1. Average length of vehicles for different countries in the ZOD.

2. Experiments

2.1. Implementation Details

In the following, we summarize some implementation details which are shared across the object detection models.

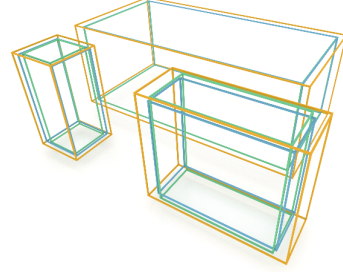


Figure 2. Comparison of average object sizes for the classes *Car*, *Pedestrian* and *Cyclist* for ZOD (green), Truck dataset (blue) the Rooftop dataset (orange). Object sizes of the Rooftop dataset are significantly larger on average.

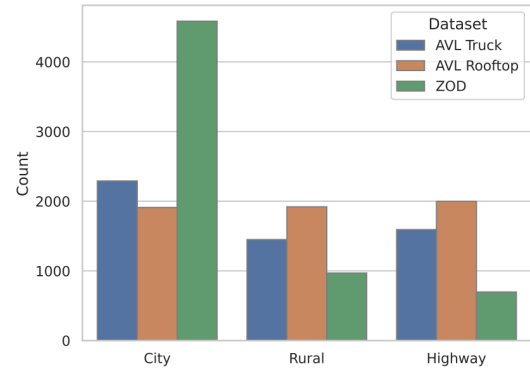


Figure 3. Recording area statistics.

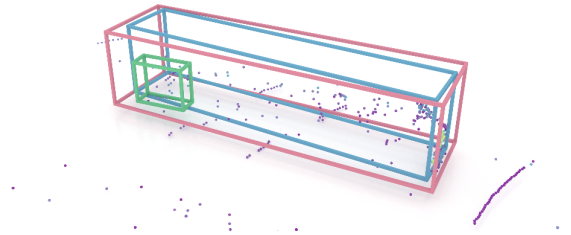


Figure 4. Example of a misclassification of an ambiguous object on a highway. The detector trained on the ZOD (green bounding box) is more likely to assign the class *Cyclist* to the ambiguous object compared to the detector trained on the Rooftop dataset (blue bounding box), which correctly identifies the object as *Truck* (red bounding box).

Codebase. All models were implemented in the codebase 3DTrans¹, which is an extension of the open-source

¹<https://github.com/PJLab-ADG/3DTrans>

Detector Label-Space	Dataset Label-Space		
	Truck	Rooftop	Zenseact Open Dataset
Vehicle	Vehicle_Drivable_Car Vehicle_Drivable_Van	Vehicle_Drivable_Car Vehicle_Drivable_Van	Vehicle_Car Vehicle_Van
Truck	LargeVehicle_Bus LargeVehicle_TruckCab Trailer LargeVehicle_Truck	LargeVehicle_Bus LargeVehicle_TruckCab Trailer LargeVehicle_Truck	Vehicle_Bus Vehicle_Trailer Vehicle_Truck Vehicle_TramTrain Vehicle_HeavyEquip
Cyclist	Vehicle_Ridable_Motorcycle Vehicle_Ridable_Bicycle	Vehicle_Ridable_Motorcycle Vehicle_Ridable_Bicycle	VulnerableVehicle_Motorcycle VulnerableVehicle_Bicycle
Pedestrian	Human	Human	Pedestrian
DontCare	Dont_Care Other	PPObject PPObject_Stroller PPObject_BikeTrailer Vehicle_PMD	

Table 1. Label-space mapping between the detector label-space and the dataset label-spaces.

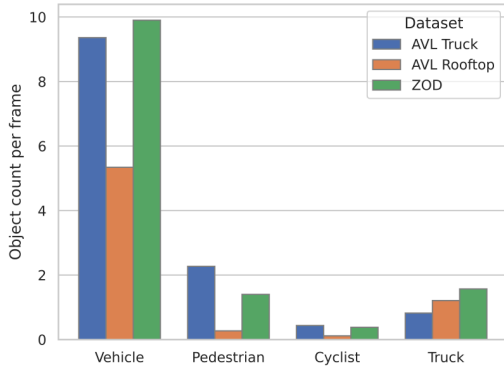


Figure 5. Class statistics.

3D object detection codebase OpenPCDet [5]. Models developed in OpenPCDet can seamlessly be integrated into 3DTrans. All the models were already implemented in 3DTrans for the Waymo Open Dataset [4], with the exception of DSVT, which had to be adopted from the official OpenPCDet codebase. The ZOD dataloader has been implemented based on the provided development kit². The dataloaders for the Rooftop and Truck datasets were implemented from scratch. Our implementations are based on PyTorch 2.1 and SpConv [1] version 2.3.6 for CUDA 12.0.

Hardware. We conducted the development and testing of the models on a workstation featuring a single RTX 4090 GPU. We trained the final models on a GPU with four RTX A6000 GPUs.

²<https://github.com/zenseact/zod>

Schedule and Optimization. We train all object detectors on each dataset for 100 epochs. All the models employ the ADAM optimizer [2] and use a OneCycle learning-rate scheduler [3] with varying learning rate, momentum and weight-decay parameters depending on the model.

Data Representation. The voxel-based methods SECOND, CenterPoint, PV-RCNN++, and DSVT require a discretization of the point cloud into a voxel-representation before the object detection models can be applied. To this end, we adapt a voxel size of (0.1m, 0.1m, 0.15m) following the implementation of PV-RCNN++. For the pillar-based method PointPillars, we use a pillar-size of (0.32m, 0.32m, 6.0m).

2.2. Detector Architecture Search

In Tab. 2, we provide the raw data used to calculate the averaged results for the domain gap and performance, which we base our detector architecture selection on. These results are also used to conduct the initial cross-domain experiment in Sec. 5.3.

2.3. Domain Gap Results

In Tab. 3 and Tab. 4 we provide the performance values in Average Precision based on which the domain gaps in Sec. 5.4 and 5.5 are calculated.

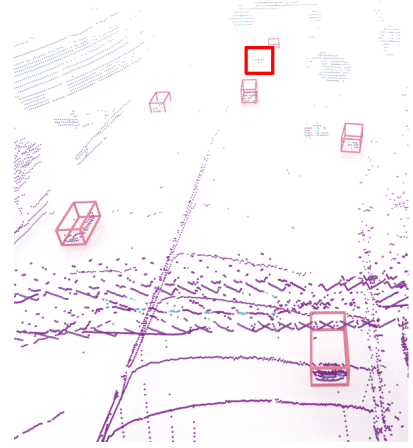


Figure 6. Example of missing ground truth label for the Rooftop dataset. The 3D bounding box of the red boxed car is missing even though it is clearly visible in the image and LiDAR data.

Source	Detector	Target		
		Truck	Rooftop	Zenseact Open Dataset
		AP ↑ IOU	AP ↑ IOU	AP ↑ IOU
Truck	SECOND	55.5/84.9	39.2/73.6	46.9/74.7
	PointPillars	49.7/82.4	32.7/67.3	33.5/67.7
	IA-SSD	58.6/82.4	41.3/69.5	53.9/72.1
	CenterPoint	54.5/82.4	36.9/75.7	43.6/74.2
	PV-RCNN++	65.5/86.4	45.5/74.0	57.6/77.4
	DSVT	60.4/86.3	41.0/72.1	53.2/77.3
Rooftop	SECOND	16.7/60.5	58.3/84.6	16.9/66.9
	PointPillars	13.6/56.3	51.8/82.3	13.6/65.1
	IA-SSD	20.6/59.9	55.1/82.2	29.9/71.7
	CenterPoint	13.9/54.4	58.3/84.0	11.7/58.9
	PV-RCNN++	21.0/58.3	61.4/84.3	22.4/73.0
	DSVT	20.9/55.2	64.6/86.4	19.9/62.6
Zenseact Open Dataset	SECOND	36.7/69.7	27.0/76.8	61.2/82.0
	PointPillars	25.3/59.5	19.1/67.6	56.7/79.8
	IA-SSD	33.7/54.0	28.7/69.3	63.8/78.6
	CenterPoint	38.7/69.8	26.9/75.5	57.7/82.7
	PV-RCNN++	44.2/68.6	32.7/75.8	68.7/84.3
	DSVT	38.1/66.2	27.6/77.0	66.4/84.3

Table 2. Detector comparison in terms of the cross-domain performance. We report the performance using the AP metric at an IOU threshold of 0.7/0.4 for the *Vehicle* class.

References

- [1] Spconv Contributors. Spconv: Spatially Sparse Convolution Library, 2022. 2
- [2] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. ICLR*, 2015. 2
- [3] Leslie N. Smith and Nicholay Topin. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates, 2018. 2
- [4] Pei Sun, Henrik Kretzschmar, Shuyang Dotiwalla, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir

Source	Target		
	ZOD ₃₂	ZOD ₆₄	ZOD ₁₂₈
	AP ↑ IOU=0.7	AP ↑ IOU=0.7	AP ↑ IOU=0.7
ZOD ₃₂	55.7	62.3	65.3
ZOD ₆₄	53.3	63.5	67.4
ZOD ₁₂₈	50.4	62.9	68.7
	AP ↑ IOU=0.4	AP ↑ IOU=0.4	AP ↑ IOU=0.4
ZOD ₃₂	70.3	77.8	82.3
ZOD ₆₄	68.4	78.3	82.9
ZOD ₁₂₈	68.2	79.0	84.3

Table 3. Density-caused domain gap for the density-resampling setting. We report the performance with the AP metric at the IOU thresholds of 0.7 (top) and 0.4 (bottom) for the *Vehicle* class.

Source	Target			
	Rooftop ₃₂	Truck ₁₂₈	Rooftop ₃₂	Truck ₁₂₈
	AP ↑ IOU=0.7	AP ↑ IOU=0.7	AP ↑ IOU=0.4	AP ↑ IOU=0.4
ZOD ₃₂	37.9	38.3	73.3	62.2
ZOD ₆₄	36.1	41.1	74.9	65.9
ZOD ₁₂₈	33.3	44.2	76.6	68.7

Table 4. Training domain gap caused by varying beam densities in a cross-domain setting. We report cross-domain performance with the AP metric at the IOU thresholds of 0.7 (left) and 0.4 (right) for the *Vehicle* class.

- Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. CVPR*, 2020. 2
- [5] OpenPCDet Development Team. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds, 2020. 2

Human Pose-Constrained UV Map Estimation

Matej Suchanek, Miroslav Purkrabek, Jiri Matas

Visual Recognition Group
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
sucham11@fel.cvut.cz

Abstract

UV map estimation is used in computer vision for detailed analysis of human posture or activity. Previous methods assign pixels to body model vertices by comparing pixel descriptors independently, without enforcing global coherence or plausibility in the UV map. We propose Pose-Constrained Continuous Surface Embeddings (PC-CSE), which integrates estimated 2D human pose into the pixel-to-vertex assignment process. The pose provides global anatomical constraints, ensuring that UV maps remain coherent while preserving local precision. Evaluation on DensePose COCO demonstrates consistent improvement, regardless of the chosen 2D human pose model. Whole-body poses offer better constraints by incorporating additional details about the hands and feet. Conditioning UV maps with human pose reduces invalid mappings and enhances anatomical plausibility. In addition, we highlight inconsistencies in the ground-truth annotations.

1. Introduction

Analysis of human pose is an essential part of many computer vision problems and is used in a number of applications, including recognition of human activity, gestures and interaction, detection of people and their intent in autonomous driving scenarios, etc.

Information about the human body can be estimated at different levels of resolution. The simplest is the detection of a bounding box that surrounds the person depicted. This can be more precisely delineated by body segmentation. *Pose estimation*, which estimates the locations of some body keypoints, provides another level of granularity. The most detailed is provided by *UV map estimation* (UVME), where every image pixel is mapped to the surface of a generalized human body. The surface is represented as a mesh with a fixed set of vertices.

The state-of-the-art methods for these tasks [10, 20, 27] rely on supervised learning, which possibly requires a large amount of annotated data. The cost and effort to annotate the data for human detection, segmentation, pose



Figure 1. The Continuous Surface Embedding method (CSE) [20] (left) vs. Pose-Constrained CSE (right). The CSE method assigns each pixel of body segmentation to a vertex, and thus UV coordinate, on a canonical body shape mesh. The CSE assigns each pixel independently, leading to artifacts such as limb duplication (yellow circles). PC-CSE uses pose constraints during UV map estimation, producing smoother maps and eliminating artifacts. The UV values at individual pixels are visualized by color coding. The location of a given color on the canonical surface is shown in the inset image at the top left.

estimation, and UV map estimation increases with the complexity of the underlying task. UVME is arguably the most complex of these tasks and, therefore, the most data-hungry.

In a recent paper, a method for UVME called Continuous Surface Embeddings (CSE) was introduced [20]. The accuracy of the method is good, but it also has limitations. Due to the disparity between the resolution of the input image and the relatively small number of vertices, this method cannot perform one-to-one matching. Since each pixel is mapped independently of the others, the method can assign the same body part to multiple locations in the image or produce undesirable artifacts. Examples can be seen in Fig. 1 and 3.

In this paper, our objective is to leverage the methods for pose estimation, which have been in development for a considerable amount of time, to make UVME more accurate. We take advantage of their robustness and design, which guarantees no duplicate assignments. We introduce the concept of *pose-induced proximal regions* which constrain the mapping to a particular body part and propagate these constraints to the corresponding pixels.

We present a novel method called Pose-Constrained CSE (PC-CSE) that demonstrates the effectiveness of these concepts. It makes UV maps more coherent with essentially no loss of efficiency besides the need to calculate the human pose. PC-CSE shows consistent improvement over unconstrained UV maps. We conducted a detailed ablation study to justify our design choices and explain the improvement in performance.

2. Related Work

Human Pose Estimation (HPE) and UV Map Estimation (UVME) are closely related tasks. UVME provides more detailed and comprehensive information, while HPE benefits from a longer history of research, larger datasets, and greater robustness. In this work, we condition UVME predictions on HPE due to HPE’s superior reliability. To establish context, we first discuss related work on HPE before moving to UVME advancements.

Data. Progress in human pose and gesture understanding relies heavily on large-scale datasets. The COCO dataset [16], with over 200,000 annotated images of people, is the most widely used, supporting tasks like object detection, instance segmentation, and pose estimation. Its annotations have been extended to whole-body keypoints [12] and UV map annotations [8]. Other datasets, such as MPII [3], CrowdPose [15], and OCHuman [28], target specific challenges like crowded scenes or people in close proximity. While these datasets have significantly advanced research, there is limited research on their overall annotation quality [22].

Current **2D Human Pose Estimation (HPE)** methods are categorized into top-down, bottom-up, and hybrid approaches. Top-down methods [18, 23, 27] first detect individuals using off-the-shelf person detectors, followed by pose estimation for each detected instance. ViTPose [27] represents the state-of-the-art in this category. Bottom-up methods [4, 6, 21] predict all keypoints simultaneously and group them into individual poses, making them more effective in crowded scenarios, such as those encountered in OCHuman [28]. Hybrid approaches [29] combine elements of both strategies, striking a balance between accuracy and efficiency under challenging conditions.

UV Map Estimation (UVME) has seen steady progress in recent years. DenseReg [7] formulates UVME as a regression task and trains a fully convolutional neural network for human face extraction using facial landmarks. DensePose [8], a milestone in UVME, collects a dataset of many body-to-surface annotations and adapts the Mask R-CNN architecture [9] for person detection, segmentation and UV map estimation in a cascade. Subsequent works focus on seeking correspondences in sequences of images [19, 24], utilize DensePose as an intermediate representation for other advanced tasks, such as 3D body reconstruction [2, 14], or use it as the ground truth [11].

DensePose relies on splitting the body template into small partitions (“charts”) and performs a simultaneous regression of the target body part and the UV coordi-

nate within the respective partition. Continuous Surface Embeddings (CSE) [20] follows up on DensePose by eliminating the need for artificial slicing of the template. Instead, CSE holds trainable descriptors (embeddings) of the template surface and guides a neural network to regress these embeddings per pixel in a contrastive manner. The UV map is determined by finding the closest surface embedding of every pixel. Overall, CSE simplifies the DensePose framework while making it generalizable to other natural objects. Both DensePose and CSE are tightly bound to the mesh of the SMPL [17], a parametrized 3D model of the human body.

BodyMap [10] further refines CSE by addressing body details such as hair and clothing, providing high-fidelity results while relying on CSE descriptors internally. Although it claims state-of-the-art performance, its code has not been released to the public. Recently, foundational models like Sapiens [13] have emerged in human-centric vision tasks. Trained on vast amounts of unannotated data, these models achieve state-of-the-art performance across various downstream tasks. However, they are resource-intensive and have yet to demonstrate significant advancements, specifically in UV map estimation.

3. Method

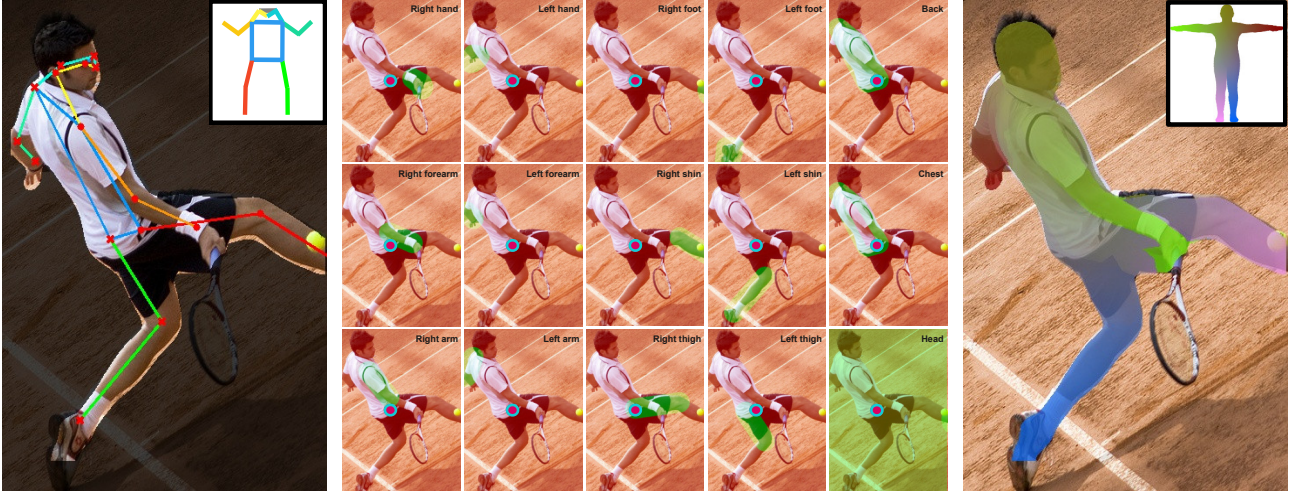
Our method is built on top of the CSE method [20], a feed-forward neural network based on the Mask R-CNN architecture [9]. Although it performs human detection, segmentation and UV map estimation in a cascade, we are concerned only with the latter and consider bounding boxes and segmentation as input determined by an external method.

The network outputs pixel descriptors, or *pixel embeddings*. During training, contrastive learning is employed to determine both the best weights and the values of *vertex embeddings*, each linked to one of the vertices of the SMPL mesh [17]. The resulting UV map is established by mapping every input pixel embedding to the most similar vertex embedding (in terms of cosine similarity), associating every image pixel with a mesh vertex (and its UV coordinates).

Formally, let I be the input image, $x \in I$ a (foreground) image pixel, $\Phi_x(I) \in \mathbb{R}^D$ embedding of the pixel x provided by the neural network Φ (where D is the embedding dimensionality), M the mesh (set of vertices), $i \in M$ a vertex index, and $E_i \in \mathbb{R}^D$ normalized embedding of the vertex i . The mapping from pixels to vertices using CSE [20] can be expressed as:

$$i_x^* = \arg \max_{i \in M} \langle E_i, \Phi_x(I) \rangle. \quad (1)$$

Consistent with the standard definition of mapping, CSE always maps exactly one vertex to every foreground pixel. However, the reverse is not necessarily true. Typically, the resolution of the input image is sufficiently high such that the pixel count significantly surpasses the vertex count on the mesh, resulting in multiple pixels being



(a) PC-CSE requires a bounding box and a segmentation mask as an input. VitPose-1 [27] is used for pose estimation in this example. Front-view skeleton is in the inset image. (b) Proximal regions of body parts. Only pixels in the segmentation mask and the green areas may be assigned to the body parts denoted in the top right. Head and undetected body parts are unconstrained (bottom right). The highlighted point can be assigned to the back, chest, head, and both the left and the right thigh but not to other parts. (c) The PC-CSE UV map is consistent with the estimated pose, unlike the CSE [20] estimate. The difference is shown in Fig. 3.

Figure 2. Pose-constrained CSE (PC-CSE) takes an estimated bounding box, segmentation mask, and 2D human pose (a) as input. It computes proximal regions (b) for each body part and assigns pixels to SMPL [17] vertices to generate a UV map. Unlike the CSE [20], PC-CSE constrains pixel assignments using proximal regions, ensuring the resulting UV map aligns with the estimated pose (c).

frequently associated with the same vertex.

Arguably, this does not pose a problem in itself. For example, it is acceptable for neighboring pixels to map to the same vertex, as they can lie so close to each other on the actual body that the discretization of the mesh cannot distinguish between them. Nevertheless, a fully independent assignment of vertices to foreground pixels makes CSE generally prone to implausible pose predictions, as it has no other means to avoid them but to rely on the strength of its prior. Qualitative research confirms our hypothesis, as we observe situations stemming from the general problem, such as CSE assigning the same body part to more than one image region (*e.g.*, two hands are declared left), UV map discontinuities and various artifacts (see Fig. 3).

At this point, we examine the features of human pose estimation (HPE) algorithms. These estimators predict the locations of various landmarks on the human body, called *keypoints*, such as skeletal joints or facial landmarks. In particular, skeletal joints form a primitive human skeleton, the shape of which is very similar to that of our 3D human representation (Fig. 2a). Furthermore, each keypoint is, by design, assigned to at most one image coordinate. This constitutes the key advantage of HPE over CSE, as duplicate assignments of body parts become impossible.

3.1. Conditioning CSE by pose

We believe that using a human pose estimation model as a secondary expert during inference and enforcing consistency of the two representations is a promising path for avoiding errors in predicted UV maps and improving their quality. Therefore, we propose our new method called **Pose-Constrained Continuous Surface Embed-**

dings (PC-CSE). The key enhancement is the introduction of *pose-induced constraints* whose purpose is to limit the mapping of every pixel to only pre-selected body regions. It does not involve any architectural change to CSE and does not require its retraining or fine-tuning.

The constraints are rules that determine to which vertices of the mesh each foreground pixel is allowed to map. Which pixels are constrained by which rule depends on the inferred pose. We first define the relation between the pose representation and the target mesh. We use the COCO skeleton [16] as the default pose representation. It consists of 17 keypoints (Fig. 2a): 12 skeletal joints (wrists, elbows, shoulders, hips, knees, and ankles in pairs) and 5 facial landmarks (eyes, ears, and nose). These keypoints can be linked into arms, forearms, thighs, shins, and a quadrilateral defined by shoulders and hips. We refer to these connections as the *principal bones*.

In addition, we explore the *whole-body* skeleton [12]. This representation with 133 keypoints extends the COCO skeleton by introducing extra keypoints for hands, feet, and face. This poses an advantage over the basic version because hands and feet are somewhat distant from the respective keypoints and can deviate from the limb axis.

The canonical mesh can now be partitioned into subsets of vertices. Each partition should roughly correspond to one principal bone. We create 15 mesh partitions of SMPL – arms, forearms, hands, thighs, shins, and feet in pairs, the front and back of the torso, and head – by merging segments of SMPL body segmentation [1, 17]. We divide the torso by the sagittal plane to distinguish between the front and back of it.

The scope of constraints within the image is specified

by expanding (“inflating”) the inferred skeleton composed of the principal bones. Each principal bone delineates its *proximal region*, each defined as a set of pixels with a certain maximum pixel distance from the bone (Fig. 2b). The optimal distance obviously relies on the apparent size of the person (which varies with its distance from the camera) and needs to be determined for each person separately. We try to estimate it using an algorithm that also depends on the pose; it is described in detail in Sec. 3.2.

The capsular shape of the proximal regions is most appropriate for the limbs, *i.e.*, arms, forearms, thighs, and shins. Concerning the front and back of the torso, we first merge the central quadrilateral (*i.e.*, between the shoulders and hips) with the regions around its sides, which we also define as having a capsule-like shape. Then, we analyze the mutual position of its corners to discriminate between the frontal and dorsal view. If the orientation of the key-points implies the frontal view of the person, we subtract the quadrilateral from the back, and vice versa (see the rightmost column of Fig. 2b).

Nonetheless, the basic COCO skeleton does not adequately support precise localization of the hands (fingers) and feet (toes). Various strategies can be employed to manage this. With the whole-body skeleton, the proximal regions for these body parts can span the extra key-points. As a fallback when using the basic skeleton, we propose circular proximal regions around the closest key-point (wrist for hands, ankle for feet) twice as wide as the capsular ones. Both these options are discussed in the experiments (Sec. 5). Otherwise, a conservative approach is to merge body parts with the nearest bone or leave them unconstrained, but this does not fully leverage the capabilities of our method. In addition, we do not outline a dedicated proximal region for the head, but we let all pixels map to it. We consider the head to be easily recognizable, and our primary goal is to resolve duplications between paired limbs.

The proximal regions induce semantic labeling of image pixels by template partitions. Every pixel is labeled according to the proximal regions to which it belongs. If multiple proximal regions overlap, the pixels within the intersection are labeled with all corresponding labels. If a pixel falls outside all proximal regions, it gets all possible labels (thus, it keeps the original prediction). When a body part is missing (that is, either of its keypoints is not provided by the HPE model), we allow mapping to it from any foreground pixel. The purpose of this rule is to prevent inaccurate refinements where, for example, a forearm is partially visible, but one of its ends lies outside the image. As described earlier, we always apply this rule to the head as well.

As a result, each pixel receives information about its target body part(s) implied by the pose-induced constraints and the embedding provided by the original CSE. We now modify the original procedure (Eq. (1)) to consider the constraints as well. Instead of yielding the vertex with the highest similarity of all mesh vertices, we limit

the output space to one of those vertices that belong to the body partitions defined by the constraints. The chosen vertex (its embedding) should still have the highest similarity to the pixel embedding, but only vertices from the limited subset of the whole mesh should be considered.

Formally, let $p \in P$ be the partition label (index), $M_p \subset M$ the vertices of the partition p , $L: I \rightarrow \mathcal{P}(P) \setminus \{\emptyset\}$ a function mapping a pixel to a set of allowed partitions. Equation (1) now becomes:

$$i_x^* = \arg \max_{i \in M_{L(x)}} \langle E_i, \Phi_x(I) \rangle, \quad (2)$$

where

$$M_{L(x)} = \bigcup_{p \in L(x)} M_p. \quad (3)$$

Alternatively, let $B(x, p)$ be the binary flag (0 or 1) indicating whether partition p is allowed in pixel x , $V(x, p)$ the vertex from partition p with the highest similarity to pixel x , $S(x, p)$ the similarity of vertex $V(x, p)$ to pixel x and $S'(x, p)$ our adjusted similarity. We compute these matrices as follows:

$$B(x, p) = \llbracket p \in L(x) \rrbracket, \quad (4)$$

$$V(x, p) = \arg \max_{i \in M_p} \langle E_i, \Phi_x(I) \rangle, \quad (5)$$

$$S(x, p) = \max_{i \in M_p} \langle E_i, \Phi_x(I) \rangle, \quad (6)$$

$$S' = S \odot B. \quad (7)$$

Equation (2) is then equivalent to:

$$i_x^* = V(x, \arg \max_{p \in P} S'(x, p)). \quad (8)$$

We believe that this approach is more practical for implementation as it avoids computing unions of mesh partitions (Eq. (3)) and storing them in memory.

3.2. Determining proximal regions

As an intermediate step, PC-CSE expands the inferred human skeleton so that its shape approximately matches the silhouette (segmentation) of the person. The exact expansion range is a trade-off. Small proximal regions might not adjust the UV map at full width. Large proximal regions can cause significant overlaps with each other, making pose-induced constraints less effective. In extreme cases, the expansion range can be chosen as zero, resulting in no correction made, or it can be chosen so high that every proximal region covers the whole body. We note that in both cases, the new prediction would be the same as, and thus *not worse than*, the original prediction.

The expansion range should roughly correspond to the width (thickness) of the person’s limbs, expressed in pixel units. We further refer to it as the *bone width* (Δ) and assume that it is proportional to other measures of the body, in particular the person’s height. Typically, information about body measures is accessible only in controlled environments where the camera model and relative location of

the object and camera are known. However, this requirement would significantly limit our method and render it useless for data “in the wild”.

Thus, we introduce a technique for estimating these measures based only on information about the person’s pose. The prerequisite is knowledge of the actual (3D) lengths of the principal bones determined by the pose estimation model. We obtain these distances from SMPL [17]. During inference, we measure the distances in the pixel space and normalize (divide) them by their distance in the 3D space. Each measurement serves as an estimate of one SMPL model unit length in pixels, assuming that the bone is parallel to the projection plane.

We then apply simple trigonometry-based reasoning to choose the most credible estimate. Given a straight unit-length stick parallel to the ground plane, its apparent length is maximal when it is parallel to the projection plane, too, and decreases when rotating the stick around the vertical axis (down to zero when both ends visually merge to the same point). In our domain, sticks are the principal bones of different lengths. Normalizing the distances by the respective lengths makes the estimates proportional only to the cosine of the angle with the projective plane. Since cosine is a decreasing function of angle (for $\alpha \in [0^\circ, 90^\circ]$), the bone having the smallest angle (ideally zero) with the projection plane will correspond to the highest value. Therefore, the best estimate is the *maximum*.

Arguably, this estimate cannot be considered perfect since we have no guarantee that the assumption of parallelism actually holds. However, we are interested in determining the size of proximal regions, which do not need to match the shape of the person exactly. In fact, a minor overestimation of the size is not a problem because we do not deal with pixels in the background anyway, and it can also help us handle people with different body mass.

Therefore, we determine the best multiplication factor by tuning it using the validation data. The results are presented in the ablation study (Sec. 5.3).

4. Data

In our experiments, we rely on the DensePose COCO dataset [8]. This dataset contains about 50 thousand annotated people on a subset of images from the COCO dataset [16]. In addition to the bounding box coordinates, instance segmentation mask, and keypoints (skeleton), the ground-truth information about every instance includes the body segmentation mask and a set of dense correspondences (over 5 million annotated points in total).

The dataset is divided into train and validation splits with a ratio of about 95/5.

4.1. Assessing the quality of annotations

During our research, we repeatedly encountered incorrectly annotated instances in DensePose COCO. Therefore, as part of our efforts, we conducted research on their

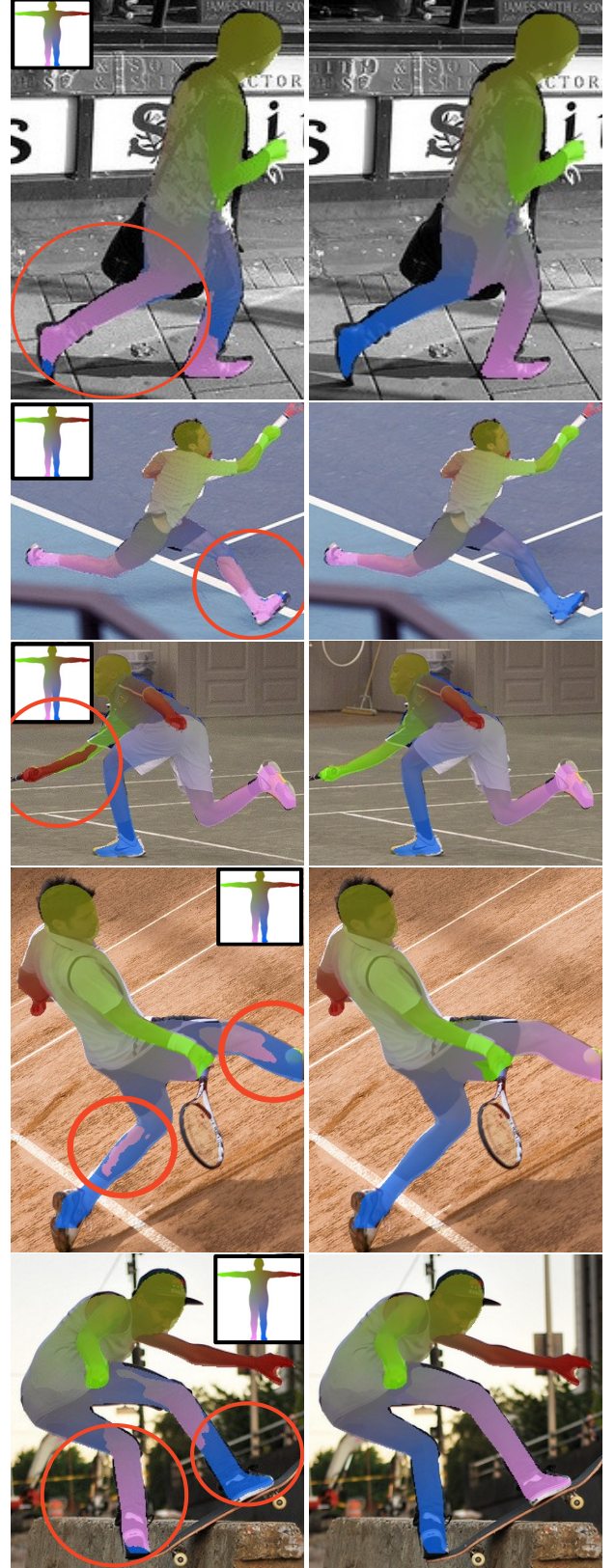


Figure 3. CSE [20] (left) vs. PC-CSE conditioned by estimated pose (right). Pose constraints ensure smoother UV maps and prevent limb duplication within a single image. A frontal view of the SMPL model [17] is shown to help assess the UV estimation.

overall quality. We define miscellaneous metrics that express the consistency of an instance’s ground truth data. (For more details, see supplementary.) Then, we manually inspect the lowest-ranking instances and identify the most common problems:

1. Annotators of dense correspondences confuse the left and right parts of the body. In most cases, only one pair of body parts is confused, while the rest are annotated correctly.
2. Dense correspondences of thighs and shins are even more confused. Some instances are annotated as having only the left or only the right leg, or annotations of one leg have mixed laterality.
3. Keypoint annotators more often confuse left and right per limb or the orientation of the entire body rather than a single pair of keypoints.
4. When multiple people at least partially overlap with a bounding box, the annotated instance is different from the one that matches the dimensions of the bounding box.
5. Body segmentation masks are incomplete; not all body parts are segmented.
6. Bounding boxes lack the “is crowd” label. These are supposed to annotate many people at once (*i.e.*, a crowd) and should not be associated with dense or keypoint annotations.

We do not make any corrections to the ground truth, but we remove dense annotations that we consider wrong. We assess the precision per body part, not individually per point. If a body part shows any of the above problems, we remove all associated points regardless of laterality. As a result, we remove ca. 1.5% points from the dataset, concerning ca. 7.5% instances. (For the validation subset, the numbers are somewhat higher: 2.4% points on 11.2% instances.)

5. Experiments

In the following, we evaluate PC-CSE by simulating its use in practice. We take the `R_101_FPN_DL_soft_slx` CSE model from the detectron2 toolbox [26] and consider it to be the baseline method. We run inference on images from the validation subset of the DensePose COCO dataset (Sec. 4) and obtain the baseline bounding boxes, instance segmentation, and pixel embeddings.

Then, we use the bounding boxes as input for top-down HPE models, which we obtain from the mmpose toolbox [5]. We choose several HPE models that differ in performance and provide different representations of human pose (see Sec. 3.1). Finally, we combine all outputs and apply our PC-CSE method and compare the accuracy of the newly produced UV maps to that of the baseline ones.

5.1. Evaluation metrics

We follow the modified COCO challenge protocol [16] that evaluates the match between predictions and ground-truth instances using Geodesic Point Similarity (GPS)

HPE method	HPE	UV map	UV map [†]
<i>None</i>	—	66.2	68.8
ViTPose-b [27]	75.8	66.8	69.3
ViTPose-h [27]	79.1	67.0	69.6
ViTPose-h wb	78.6	67.3	69.8
RTMPose-l [18]	75.8	67.0	69.5
RTMPose-l wb [18]	69.5	66.7	69.3

Table 1. **AP results on the COCO dataset.** Constraining UV map estimation with 2D pose improves performance. More accurate poses lead to better UV maps. Using the whole-body (wb) skeleton further enhances performance due to better hand and foot constraints. Note that 2D Human Pose Estimation (HPE) is evaluated on a different COCO subset than UV map evaluation. Results marked with ([†]) are evaluated on data with ignored incorrect annotations, as detailed in Sec. 4.1.

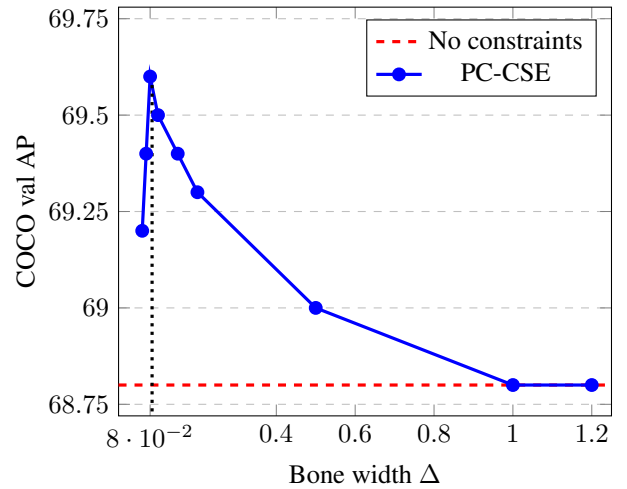


Figure 4. **Ablation on bone width Δ** defined in Sec. 3.2. RTMPose-l wb [18] is used for pose constraints. Too thin bones restrict UV Map too much and hinder performance on border pixels. Excessively thick bone estimates do not restrict UV Map sufficiently and reduce the performance gain. Note that performance with proximal regions with large regions Δ converges to the baseline method. In the extreme case when all bones are as big as the whole picture, no constraints are applied. The best value is 0.08.

and computes the algorithm’s Average Precision (AP) by thresholding the GPS score [8]. We report the Average Precision for both the original dataset and the dataset without incorrect annotations (see Sec. 4.1).

5.2. Results

Table 1 compares pose-constrained CSE (PC-CSE) with the original CSE [20]. The results are reported in the COCO val dataset for comparability with previous work. Furthermore, we evaluated performance on the COCO val data set while ignoring incorrect annotations, as described in Sec. 4.1.

The first row of Tab. 1 shows the performance of CSE [20] without pose constraints. We reproduced these re-

sults and observed a 2.6 AP improvement when ignoring incorrect annotations. This gain remains consistent across all experiments.

Subsequent rows show results with pose constraints from ViTPose [27] and RTMPose [18], using different model variants. Regardless of the HPE model, applying pose-conditioned constraints consistently improves performance. As expected, the performance gain depends on the quality of the HPE model. ViTPose-h (huge) outperforms ViTPose-b (base) in HPE and achieves slightly better UV map accuracy. However, the difference is minor. Note that HPE is evaluated on a larger subset of COCO images than UV maps.

To assess the impact of the whole-body (wb) skeleton, we trained *ViTPose-h wb* on the COCO-WholeBody dataset [12]. It achieves 67.3 AP on COCO-WholeBody and 78.6 AP on COCO, compared to 79.1 AP for ViTPose-h. While the whole-body poses are less accurate, the inclusion of fingers and toes compensates for this in specific body regions.

Results for RTMPose [18] follow a similar trend. Using estimated poses improves the performance of the UV map between models, although exact gains differ. For instance, RTMPose-l matches ViTPose-b in HPE performance, but achieves slightly higher UV map accuracy. However, this difference is negligible.

RTMPose-l wb shows a much weaker HPE performance but comparable UV map accuracy. Although the inclusion of fingers and toes benefits the hand and foot regions, the reduced accuracy of other keypoints diminishes overall gains, making the trade-off less favorable.

While conditioning UV map predictions on pose significantly improves consistency, this translates to only a modest 1 AP point increase in overall performance due to several factors. The most significant issue is segmentation errors — pixels outside the segmentation mask are not assigned UV map estimates, leading to penalties. An example is shown in image Fig. 6. Detection errors also impact performance; if a person is not detected, no UV estimation can be performed.

Achieving 100 AP is challenging due to the limitations of ground truth annotations, which are human estimates often obscured by clothing. In images with loose clothing, these annotations can be highly imprecise, making it difficult to determine whether discrepancies stem from ground truth errors or model predictions. As a result, images with GPS around 80 already represent strong estimates, as shown in Fig. 6.

Examples of significant improvements over the baseline are shown in Fig. 1 and Fig. 3. These include artifact removal, better continuity between limbs, and elimination of redundant body part assignments in baseline UV maps.

5.3. Ablation study

The efficiency of PC-CSE depends on a proper outline of the proximal regions, as described in Sec. 3.2. To ensure overall robustness, we determine the best value of

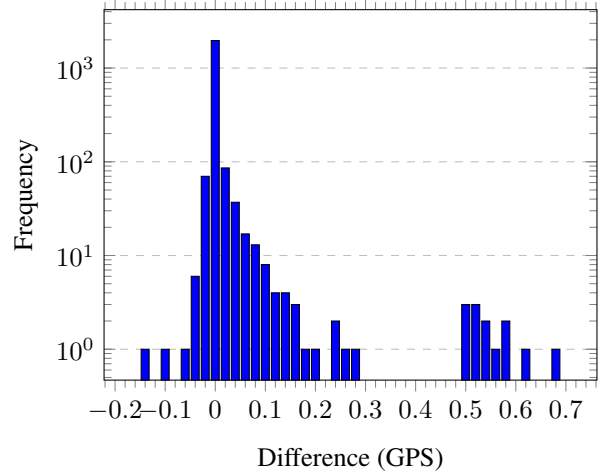


Figure 5. Image-wise performance difference of baseline CSE [20] and PC-CSE with poses from ViTPose-h wb. Performance is measured by GPS, frequency is in log scale. Positive means better performance of PC-CSE. PC-CSE causes only a few performance drops while improving more other cases, some of them dramatically, and keeping the rest about the same.

the bone width hyperparameter Δ by validation. We use the RTMPose-l wb model [18] and repeat the same experiment while varying the value of the hyperparameter. Note that we use the clean validation dataset that does not contain the incorrect annotations identified (Sec. 4.1).

The results, shown in Fig. 4, confirm our expectations. With an increasing value of the hyperparameter, the precision increases and reaches the maximum when it is equal to 0.08. Increasing it further, we observe a gradual decrease in precision down to the baseline. This supports our earlier statement (Sec. 3.2) about the best value being a compromise and the consequences arising from a suboptimal choice. Extremely small and large values do not give our method the opportunity to have the desired impact.

Note that our experiments generally assume that the method for estimating a person’s measures (Sec. 3.2) from their pose is accurate. We do not conduct any quantitative experiments on this matter, but we attempt to verify it using qualitative analysis (see supplementary material).

In addition, we provide a detailed analysis of the variation in performance metrics for each evaluated sample. An example histogram, generated for ViTPose-h wb, is shown in Fig. 5. We notice that the model maintains baseline precision on the vast majority of data samples and observe only a few performance drops, which are mainly caused by failure in the underlying pose inference. The worst are depicted in Fig. 6. However, these failures are largely compensated for by more common, sometimes drastic, improvements (shown in Fig. 3).

6. Conclusions

We presented Pose-Constrained CSE (PC-CSE), a method that conditions UV map estimation using human pose. PC-CSE leverages the robustness of 2D human pose es-

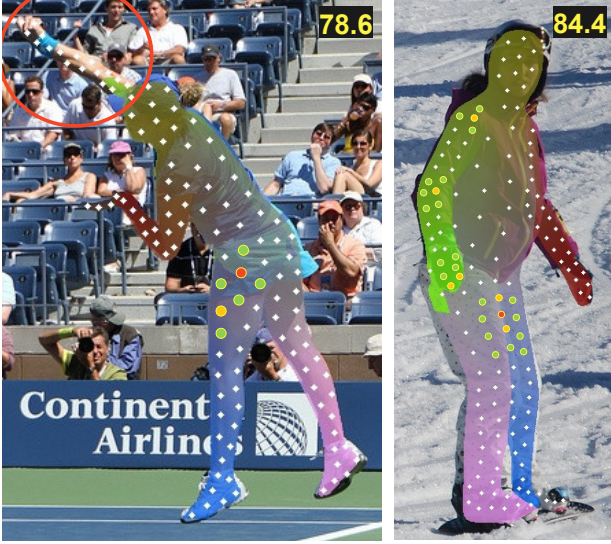


Figure 6. Images with GPS (geodesic point similarity) around 0.8. Evaluation points are shown in white. Selected wrongly estimated points (similarity < 0.5), slightly wrong (similarity 0.5 – 0.9), and correct (similarity > 0.9). Typical errors are isolated wrong points among correct ones (left, hip), segmentation errors (left, red circle), and border points (right, legs). Loose clothing complicates annotation and estimation (right).

timination to provide global constraints, improving the consistency of UV map predictions produced by CSE [20].

The original CSE [20] assigns pixels to vertices independently, which can lead to errors, such as assigning the same body part to multiple locations in the image and discontinuities in the same body part, as shown in Fig. 3. PC-CSE introduces global supervision through pose constraints, ensuring that while pixel assignments remain independent, the global pose structure improves the consistency of the UV map. This results in more coherent UV maps, free from artifacts and duplicated limbs.

Key findings are:

1. Conditioning UV maps by pose, even with rudimentary constraints, provides consistent improvements, though overall performance gains remain modest.
2. The choice of pose estimation model architecture has a negligible impact on the results.
3. Whole-body skeletons enable more precise constraints for hands and feet, yielding small improvements over body-only skeletons without additional computational costs.
4. COCO DensePose annotations are not entirely reliable; at least 1.5% of the points are inconsistent with pose keypoints or are otherwise inaccurate. The accuracy of points under loose clothing remains uncertain as we could neither confirm nor disprove their precision.

Limitations. The primary limitation of PC-CSE lies in its reliance on precise pose estimation. The method assumes that 2D human pose estimation (HPE) models are robust to challenges such as extreme poses, occlusions,

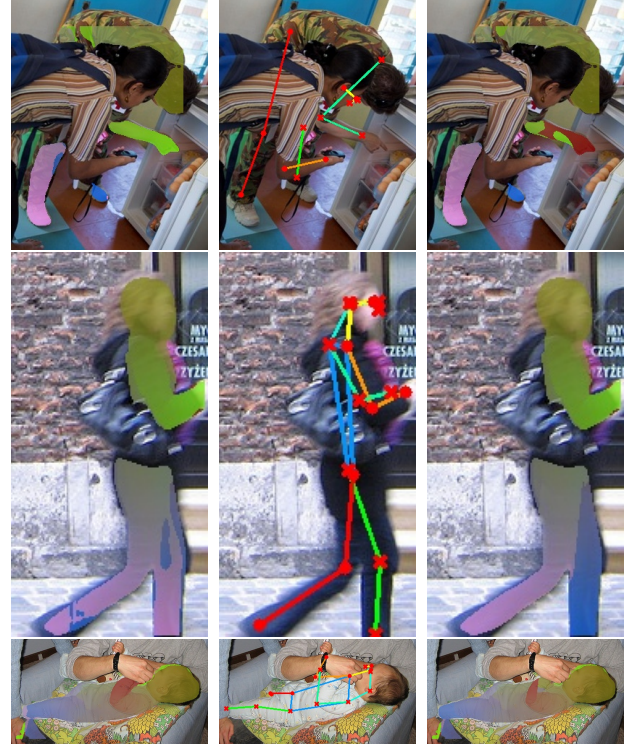


Figure 7. Three images with the largest performance decrease – CSE (left), pose estimate (middle), PC-CSE (right). Pose conditioning reduces performance when the pose estimation fails. Despite the drop, the third most negatively affected image (bottom) shows only a 0.5% decrease, highlighting that pose conditioning negatively impacts only a few images while improving many others.

and image deformations, which can condition UV map estimation effectively. However, if the estimated pose is inaccurate, the constrained UV map will also be incorrect. The most common errors occur in multi-body scenarios.

Another limitation arises when two body parts are in close proximity. For instance, when a person is sitting with crossed legs, pose constraints for both legs might overlap, preventing PC-CSE from correcting the original CSE estimates. Although PC-CSE does not resolve such issues, it does not degrade overall performance.

Future work. The constraints implemented by us are very coarse, as they are satisfied by letting the pixel map *somewhere* on the given body part. The corrections could become even more precise by taking the distance from its endpoints (keypoints) or the orientation of the body (frontal/dorsal) into account. In addition, there is substantial redundancy in the HPE and CSE representations, while the HPE algorithms are more advanced. The CSE method could be redesigned by building it on top of HPE and changing its objective to provide UV map estimation *given a pose estimate* (and not just the image). We also plan to use the method for UV maps on animals using SMAL [30].

Acknowledgements. This work was supported by the Ministry of the Interior of the Czech Republic project No. VJ02010041 and Czech Technical University student grant SGS23/173/OHK3/3T/13.

References

- [1] SMPL - Meshcapade Wiki — meshcapade.wiki. <https://meshcapade.wiki/SMPL>. [Accessed 2024-12-22]. 3
- [2] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus A. Magnor. Tex2shape: Detailed full human body geometry from a single image. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2293–2303, 2019. 2
- [3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2
- [4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 2
- [5] MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. <https://github.com/open-mmlab/mmpose>, 2020. 6
- [6] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14676–14686, 2021. 2
- [7] Riza Alp Güler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2614–2623, 2016. 2
- [8] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 2, 5, 6, 1
- [9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 2
- [10] Anastasia Ianina, Nikolaos Sarafianos, Yuanlu Xu, Ignacio Rocco, and Tony Tung. Bodymap: Learning full-body dense correspondence map. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13276–13285, 2022. 1, 2
- [11] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12748–12757, 2021. 2, 1
- [12] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 3, 7
- [13] Rawal Khirodkar, Timur Bagautdinov, Julieta Martinez, Zhaoen Su, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. *ArXiv*, abs/2408.12569, 2024. 2
- [14] Eric-Tuan Lê, Antonis Kakolyris, Petros Koutras, Himmy Tam, Efstratios Skordos, George Papandreou, Riza Alp Güler, and Iasonas Kokkinos. Meshpose: Unifying densepose and 3d body mesh reconstruction. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2405–2414, 2024. 2
- [15] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Haoshu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10855–10864, 2018. 2
- [16] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 2, 3, 5, 6, 1
- [17] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 2, 3, 5
- [18] Peng Lu, Tao Jiang, Yining Li, Xiangtai Li, Kai Chen, and Wenming Yang. Rtmo: Towards high-performance one-stage real-time multi-person pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1491–1500, 2024. 2, 6, 7
- [19] Natalia Neverova, James Thewlis, Riza Alp Güler, Iasonas Kokkinos, and Andrea Vedaldi. Slim densepose: Thrifty learning from sparse annotations and motion cues. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10907–10915, 2019. 2
- [20] Natalia Neverova, David Novotný, Vasil Khalidov, Marc Szafraniec, Patrick Labatut, and Andrea Vedaldi. Continuous surface embeddings. *Advances in Neural Information Processing Systems*, 33:17258–17270, 2020. 1, 2, 3, 5, 6, 7, 8
- [21] George Papandreou, Tyler Lixuan Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin P. Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *European Conference on Computer Vision*, 2018. 2
- [22] Arnold Schwarz, Levente Hernadi, Felix Bießmann, and Kristian Hildebrand. The influence of faulty labels in data sets on human pose estimation. *arXiv preprint arXiv:2409.03887*, 2024. 2
- [23] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 2
- [24] Feitong Tan, Danhang Tang, Mingsong Dou, Kaiwen Guo, Rohit Pandey, Cem Keskin, Ruofei Du, Deqing Sun, Sofien Bouaziz, S. Fanello, Ping Tan, and Yinda Zhang. Humangps: Geodesic preserving feature for dense human correspondences. *2021 IEEE/CVF Conference on Computer*

- Vision and Pattern Recognition (CVPR)*, pages 1820–1830, 2021. 2
- [25] TikTok. Tiktok. <https://www.tiktok.com>. [Accessed on 2024-12-17]. 1
- [26] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. 6
- [27] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In *Advances in Neural Information Processing Systems*, 2022. 1, 2, 3, 6, 7
- [28] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul L. Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shimin Hu. Pose2seg: Detection free human instance segmentation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 889–898, 2018. 2
- [29] Mu Zhou, Lucas Stofl, Mackenzie W. Mathis, and Alexander Mathis. Rethinking pose estimation in crowds: overcoming the detection information bottleneck and ambiguity. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14643–14653, 2023. 2
- [30] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 8

Human Pose-Constrained UV Map Estimation

Supplementary Material

A. Annotation data quality assessment

In Sec. 4.1, we conduct research on the quality of annotations from the DensePose COCO dataset [8, 16]. In order to efficiently identify the majority of erroneous annotations without having to manually examine the entire dataset, we take the following approach.

We establish several metrics to quantify the level of (in)consistency or “(im)plausibility” of each annotated example:

- Proportion of the body segmentation covered by the instance mask. By definition, the body mask should be completely covered by the instance mask. This allows revealing problems 4 and 6, as enumerated in Sec. 4.1.
- Proportion of the area of the instance or body mask and the bounding box. Ideally, the mask should cover a significant portion of the bounding box.
- Proportion of point-wise annotations within the instance or body mask. Human segmentation should ideally contain all (visible) keypoint annotations and dense correspondences, which concern the body, too. Likewise, this allows revealing problems 4, 5, and 6.
- Ratio of median points-to-bone distances.

We group ground-truth dense correspondences by body part and compute their median distance to the respective bone defined by ground-truth keypoint annotations (bone selection is done analogously to our mesh partitioning procedure, which exploits its resemblance to the COCO skeleton; see Sec. 3.1). We add up median distances for the same body part of either laterality. Then, we repeat the same procedure with the laterality of the keypoints flipped, and obtain another score. The ultimate value of the metric is the ratio of the two sums. When this value is high ($\gg 1$), it indicates a possibly confused laterality of keypoints or dense correspondences.

This allows revealing problems 1 and 3.

- Inference error. We run inference on all images from the dataset and compute the mean geodesic distance (error) per body part. High inference error usually indicates deficiencies in the model’s performance, but, especially on training data, it might also help reveal annotation errors. We took advantage of repeated retraining and evaluation of the inference model (“human in the loop”) as it could initially have been overfitted to annotation errors.

This allows revealing problems 1 and 2.

We sort all annotations from the least consistent and manually examine them in this order until annotations with no apparent problems start to prevail. This process is carried out individually for each defined metric.

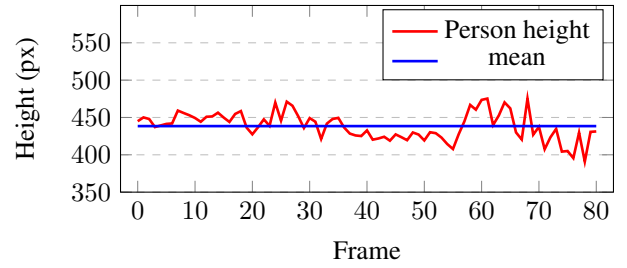


Figure 8. **Ablation on height estimation.** We infer pose from a dance video from [11] at 10 frames per second and estimate the dancing person’s height in pixels (red) using the algorithm in Sec. 3.2. The variable exhibits some noise due to pose changes, but remains within the interval of a few tens of pixels at all times. The bigger noise at the end of the video is caused by more extreme poses.

B. Ablation study on height estimation

Our PC-CSE relies on estimating the proper outline of each constraint’s region. In Sec. 3.2, we describe the algorithm we use to approximate the person’s body measurements in pixel units of the image using only its inferred pose. The precision of such an algorithm can usually be determined by comparing the actual values and their algorithmic estimate on many images. We do not conduct such an experiment because of the lack of ground-truth data, but we verify its performance by taking a different approach.

The goal is to demonstrate that the estimate is not dramatically influenced by pose variations. However, images of people “in the wild” usually also differ in the distance of the person from the camera, as well as the underlying camera parameters. For a sensible comparison, these two factors need to remain constant. We notice that this requirement is met, for example, by short videos of people dancing in front of the camera uploaded to social networks such as TikTok [25].

Therefore, we take advantage of the TikTokDataset [11] and select several videos where a person performs a dance in front of a static camera without moving around the place. We run pose inference per video frame and record the height estimate. An example chart recording the progress of one video is shown in Fig. 8. The variable does exhibit some noise, approximately on the scale of tens of pixels, which can be attributed to pose variations and noise in the pose estimation, but it remains centered on its mean value throughout. We note that the actual noise influencing the estimate of bone width (Δ) is much smaller since the bone width is a small fraction of the person’s height.

Incremental Learning with Repetition via Pseudo-Feature Projection

Benedikt Tscheschner^{1,2}

Eduardo Veas¹

Marc Masana^{2,3}

¹Know-Center Research GmbH

²Institute of Visual Computing, TU Graz

³SAL Dependable Embedded Systems, Silicon Austria Labs

{btscheschner, eveas}@know-center.at, mmasana@tugraz.at

Abstract

Incremental Learning scenarios do not always represent real-world inference use-cases, which tend to have less strict task boundaries, and exhibit repetition of common classes and concepts in their continual data stream. To better represent these use-cases, new scenarios with partial repetition and mixing of tasks are proposed, where the repetition patterns are innate to the scenario and unknown to the strategy. We investigate how exemplar-free incremental learning strategies are affected by data repetition, and we adapt a series of state-of-the-art approaches to analyse and fairly compare them under both settings. Further, we also propose a novel method (Horde), able to dynamically adjust an ensemble of self-reliant feature extractors, and align them by exploiting class repetition. Our proposed exemplar-free method achieves competitive results in the classic scenario without repetition, and state-of-the-art performance in the one with repetition.

1. Introduction

As autonomous agents and models in production systems are exposed to continuous streams of information, they are required to adapt to dynamic data distributions with potentially multiple tasks and integrate new information over time [3, 28, 43]. The practice of retraining the complete system whenever new data is available becomes unfeasible as the storage, computation and privacy constraints for data streams increase [31, 35, 39]. To address these constraints, incremental learning (IL) or continual learning has emerged as a promising approach [4].

IL aims to learn a model sequentially through a sequence of tasks introducing disjoint sets of information at each training step [8, 27, 42]. Generally, these scenarios enforce a strict no-repetition constraint [7] allowing access to the data distribution only once in the task sequence. Unlike humans, who can learn nearly inference-free between tasks, neural networks suffer from a phenomenon called *catastrophic forgetting* [10, 12]. When models are optimized sequentially on novel tasks, a swift forgetting of previously learned tasks is observed. To mitigate this

forgetting, a delicate balance between preserving learned task knowledge (stability) and the ability to adapt to new information (plasticity) has to be reached, which is known as the stability-plasticity dilemma [29]. A popular approach to address this is to cache a representative subset of previously encountered data points in a buffer and replay them during the following training sessions [38, 41, 42]. Although such *rehearsal* addresses catastrophic forgetting effectively, data privacy concerns have been raised [14], and the scalability of an exemplar buffer in long-tailed incremental sequences is questionable [42] due to the large computational cost of complete retraining and significant storage requirements.

Nonetheless, the strict enforcement of no-class repetition becomes unrealistic for many real-world applications, as continuous streams are bound to repeat certain information [7] or be affected by semantic or covariate shifts [30]. For example in industrial defect detection, certain common defects and defect-free samples will repeat throughout production. The occurrence of repetition is further amplified in environments where an agent has the freedom to reexperience elements which are contained within the overall environment design. Thus the effects of catastrophic forgetting are likely exaggerated as an uncontrollable form of rehearsal occurs naturally. Previous incremental learning research has largely explored catastrophic forgetting under the assumption that new information has a single opportunity to be learned, since each class is only available within a single task throughout the sequence. The introduction of repetition into these scenarios enables the selection of more broad incremental training tasks and highlights the different dynamics within the plasticity-stability dilemma of learning new tasks while maintaining current knowledge [7]. The focus on catastrophic forgetting without repetition may limit the development of more realistic incremental learning agents, which involve different complex objectives like forward transfer [24] and efficiency for computational limitations in edge devices [9].

As such, we want to loosen the no-repetition constraint and explore the effects of natural repetition. To explore these new settings and effects, our contributions are:

1. a new variation of the class-incremental learning

- CIFAR 50/10 scenario introducing class repetition,
2. benchmarking a broad selection of state-of-the-art exemplar-free class-incremental learning methods and investigate the effects of innate data repetition and their resiliency to repetition frequency bias,
 3. a novel incremental learning method (Horde) that builds an ensemble of independent feature extractors for stability and utilizes pseudo-feature projection for plasticity (see Fig. 1).

2. Related Work

Class-incremental learning (CIL) addresses the challenge of training a model sequentially on a series of tasks, without access to previous or future data [36]. When training without any constraints, models fail to retain knowledge from previous tasks – a problem known as *catastrophic forgetting* [10, 12]. Usually, each incremental task contains a disjoint set of new classes, which increases the difficulty of discriminating between those which have not been learned together under the same task [8]. A key challenge in incremental learning lies in keeping the balance of the stability-plasticity dilemma [29], critical for mitigating catastrophic forgetting while ensuring the adaptability of the model to new tasks.

Incremental learning approaches include: weight regularization [2, 20], which preserves important weights by estimating their importance; knowledge distillation [18, 22], which focuses on protecting task representations rather than weights; rehearsal [38], which replay stored exemplars from previous tasks; mask-based approaches [26], which use task-specific masks to isolate parameters that can be updated; and dynamic network structures [11, 32], which expand the model architecture by adding new or contracting existing modules for each task. In this work, we concentrate on weight-regularization, knowledge distillation and dynamic network structure-based methods. These are the approaches that work on task-agnostic scenarios (do not require a task-ID during inference) and promote privacy preservation (do not store samples).

Incremental learning with repetition. In many practical applications (automated failure inspection, medical imaging, robotics), pattern repetition naturally arises, yet traditional CIL approaches assume that each class is encountered only once, imposing a strict no-repetition constraint [7]. This constraint focuses on the prevention of catastrophic forgetting but also diverges from real-world scenarios where classes may reappear or shift over time. To address this, Hemati et. al [15, 16] propose an extension to the class-incremental learning scenario which models the repetition of individual classes outside of a single task. Unlike joint incremental or rehearsal-based learning, this repetition is innate to the learning scenario and cannot be adjusted. This emphasizes an experience-based scenario [41], which favours shorter training tasks that can sometimes only cover a part of the class distribution.

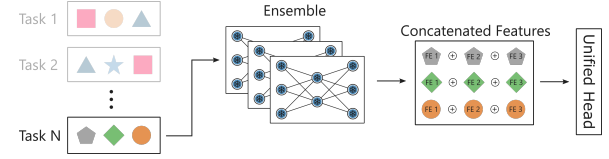


Figure 1. Overview of our proposed method (Horde). Each data sample is processed by an ensemble of independent feature extractors. The features from all extractors are concatenated before being passed into a unified head that can accommodate the dynamic input size through pseudo-feature projection.

Moreover, covering scenarios that lie between the classic offline incremental and the online ones.

Class-incremental learning with repetition has received increased interest in the research community, being a central element in the challenge tracks of the last two CLVISION challenge tracks at CVPR 2023 and 2024 [1, 16]. In the 2023 edition, we competed with a base variant of our proposed method, although without elements for controlling ensemble growth (see Sec. 3.1), self-supervision (see Sec. D) or applicability to variable network architectures.

Class prototypes and pseudo-features. To enforce stability and alleviate class-recency biases in the classifier [27], Exemplar-Free Class Incremental Learning (EFCIL) methods [33, 40, 46–48] utilize class prototypes to simulate unavailable classes. These prototypes capture statistical properties of embedding representations of each class, which are usually modeled as a multivariate Gaussian distribution [40, 46, 47]. Specifically, the statistics typically include the mean and covariance of feature representations for each class, allowing to generate pseudo-features when class data is not available. To extract representations, the neural network is divided into two modules. A feature extractor (FE) that projects the input samples into their corresponding embedding representation; and a classifier head that uses these embeddings to solve the classification task. Therefore, prototype-based methods can generate embeddings even when no samples from past classes are available during subsequent tasks by sampling the stored distributions of each class. The sampled embedding representations are rehearsed alongside the current task data, thus promoting stability and mitigating class-recency bias. However, in order to maintain valid approximations of class distributions, the feature extractor needs to be either frozen or heavily regularized to prevent changes or drifts in the extracted features. Unlike rehearsal-based approaches, the use of prototypes does not violate data privacy due to the non-linearly projected representation in the embedding space [44, 47].

Feature translation. Instead of sampling the distribution approximated by class prototypes, FeTrIL [33] proposes to translate the features of available data classes to unavailable ones directly. Given a feature extractor $f(x; \theta)$ being trained on current data $\{(x_i, y_i)\}$, its output embedding F is efficiently translated from one of the current

classes to the desired previously learned class $c \in \mathcal{Y}$ as

$$\hat{F}_c = \mathbf{f}(x_i; \theta) + \mu_c - \mu_{y_i}, \quad (1)$$

where μ_c and μ_{y_i} represent the means of the old and current classes, respectively. The feature translation modifies the classifier, however, the feature extractor is required to be frozen after the initial training so that the class means can be reliably extracted. This limits the continual learning process as the initial task constrains the diversity and robustness of the features that can be learned for new classes [4, 33]. In our proposed approach, we relax this restriction by allowing an ensemble of smaller feature extractors to be learned. This allows for unknown class prototypes to be estimated through pseudo-feature projection until the repetition of classes allows for an accurate extraction of class prototypes.

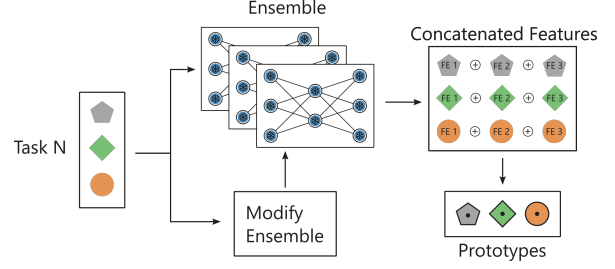
3. Method

In incremental learning scenarios with repetition, the reappearance of classes introduces uncertainty in task sequences, requiring strategies that handle dynamic class distributions. Our approach aims to: (a) capture information from the current task, (b) integrate it with knowledge from previously seen tasks, and (c) ensure the ability to discriminate between all encountered classes so far. To achieve this, we leverage zero-forgetting feature extractors (FEs), which are aggregated in an ensemble to overcome the limitation of a completely fixed feature space. Through this aggregation, we form a flexible feature representation space that can adapt (expand or contract) based on the incremental learning sequence (see Fig. 1 for an overview of the proposed method structure).

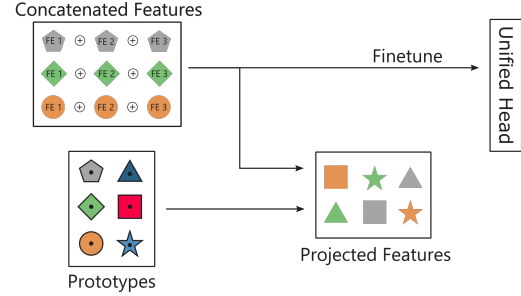
To effectively utilize this dynamic embedding space, we address the challenge of missing classes by constructing prototypes for all encountered classes. Class prototypes are used to train a unifying classification layer through an adjusted feature translation mechanism, termed *pseudo-feature projection*, ensuring continuous adaptation and robust performance across all classes. Concretely, the learning approach is divided into two steps: **(1st)** based on the difficulty of the current task and depending on how well new classes can fit into the ensemble embedding space, the ensemble is expanded with a new feature extractor (described in Sec. 3.1); **(2nd)** once the embedding space has been fixed for the current task, class prototypes are extracted and the unified classification layer is trained through the *pseudo-feature projection* (described in Sec. 3.2). These steps are performed for every incremental task and are summarized in Figure 2.

3.1. Feature Extractor Ensemble

The proposed aggregation framework consists of multiple individual feature extractors (FEs), each trained on a specific task and then frozen to preserve the learned representations. The motivation for this zero-forgetting strategy is to enforce stability, avoiding any catastrophic forgetting on the ensemble while providing some plasticity



(a) **Step 1:** The ensemble of feature extractors is adjusted based on the current task through either the addition or update of a self-reliant feature extractor. This step is only performed when estimated as necessary via a heuristic criteria.



(b) **Step 2:** Class prototypes are extracted or updated from the current task data. Incomplete class prototypes (those estimated before Step 1 extends or modifies a feature extractor) are updated and data for unavailable classes is simulated by pseudo-feature projection. An unbiased classification head is finetuned from the current training data and the projected features of unavailable classes.

Figure 2. Overview of the steps our proposed method (Horde) performs for each incremental task.

through the extension of the ensemble. Unlike FeTrIL, which freezes a single feature extractor after the initial training, the extension of the feature space through the ensemble relieves the dependence of an expressive initial feature extractor. The goal of each feature extractor is to build a diverse and expressive feature space that emphasizes high-quality representations rather than optimizing the performance of the individual incremental task. Further, we adopt the self-learning loss from PASS [47]. This self-learning loss enhances the learned feature representation by simultaneously classifying image orientation and categories (each image class now has 4 augmented labels depending on the image orientation). To further improve regularization on the feature space topology, we incorporate a metric learning head with contrastive loss [34] and hard-negative mining [37]. This promotes *spherical-shaped* clusters in the embedding space, which improves class discrimination between known and unknown distributions [25]. Additionally, the sphere-shaped structure aligns well with the properties of a multivariate Gaussian distribution, which relates to the pseudo-feature projection we propose. An ablation study of the effects of individual components is provided in the supplementary material (see Sec. D).

Ensemble Growth. To control the growth of the ensemble, we set a predefined budget B for the maximum num-

ber of FEs. For each incremental task, a decision is made whether the concatenated embedding space should be adjusted based on the following criteria:

- *constant feature representation*: when the current ensemble embedding representation is sufficient to handle the incremental task, no new FE is trained. New classes are learned using the existing ensemble representations without requiring additional feature extraction capacity.
- *dynamic feature adaption*: when the current ensemble of FEs cannot adequately represent the new task due to a significant change in the data distribution, task complexity or overlap with previous classes, a new FE is added.

To capture these criteria and guide the growth of the ensemble, we propose two heuristics to guide the modification of the ensemble (see Step 1 in Fig. 2a):

- **Class Set Maximisation (Horde_m)**: this heuristic aims to maximize the diversity of classes represented across the ensemble. Specifically, it ensures that each FE contributes to representing as much of a distinct set of classes as possible

$$\max \bigcup_{i \in B} |c \in F^i|, \quad (2)$$

thereby increasing the overall coverage of the class space across all feature extractors. This maximization is tested at the start of each incremental task. Thus, when a larger class set is possible with the current incremental task data, a new FE is trained. The new FE either is added or replaces one in the ensemble.

- **Task Error Rate (Horde_e)**: At the start of the incremental task, the error rate e on the current incremental data is computed (before training). It is obtained from the confusion matrix (CM) by calculating the ratio of wrong predictions over all other predictions:

$$e = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} \left(\frac{\sum_{j \neq c} \text{CM}_{c,j}}{\sum_i \text{CM}_{c,i}} \right). \quad (3)$$

If e is too high, the incremental data cannot be classified with the current ensemble effectively. Therefore, we introduce a threshold or budget of the ensemble B which signals the need to train a new feature extractor based on e . After training the unified head (Step 2), an *improvement score* is calculated as the difference between the error rate at Step 1 (before any training is performed) and after Step 2. If the budget B has been exceeded the FE with the lowest improvement score is replaced.

3.2. Unified Classification Layer

To unify the feature representations from the ensemble and enable task-agnostic classification, we utilize a fully-connected layer. This layer has dynamic input and output sizes depending on the growth of the ensemble and

the number of incrementally learned classes. To mitigate task-recency bias [8], we train this unified head using both data from the current task and projected class prototype features through our proposed pseudo-feature projection.

Pseudo-feature projection. Pseudo-feature projection, inspired by FeTrIL [33], extends feature translation by incorporating both the mean and standard deviation of class prototypes. This enhances the sampling of dimensions, reduces the chance of overlapping classes in the embedding space and leads to more accurate feature replay. With this projection, a data point from one class may be projected to a pseudo-feature representation of any other previously learned class. Our proposed projection extends the one from FeTrIL on Eq. (1) as

$$\hat{F}_c = \mu_c + \frac{\mathbf{f}(x_i; \theta) - \mu_{y_i}}{\sigma_{y_i}} \cdot \sigma_c, \quad (4)$$

where \hat{F}_c represents the pseudo-features of the latent representation of a data point (x_i, y_i) which is projected from the original class y_i to the desired class c . This transformation leverages the class prototypes; specifically the mean μ_{y_i} and standard deviation σ_{y_i} to modify the latent representation $\mathbf{f}(x_i; \theta)$. Class prototypes are updated during Step 2, before training the unified classification layer and after the ensemble has been adjusted.

We represent a complete class prototype as the concatenation of the individual class statistics from each FE in the ensemble:

$$\begin{aligned} \mu_c &= (\mu_{c,1}, \dots, \mu_{c,n}), \\ \sigma_c &= (\sigma_{c,1}, \dots, \sigma_{c,n}), \end{aligned} \quad (5)$$

where n determines the current size of the ensemble. Throughout the incremental sequence, the ensemble can be expanded until the feature extractor budget is exhausted ($n \leq B$). Once this limit has been reached, individual feature extractors need to be finetuned or replaced and their corresponding class prototype $(\mu_{c,i}, \sigma_{c,i})$ is reset.

Class prototypes of certain classes may be incomplete for newly added or modified FEs. When class statistics are unknown for a specific FE, estimates are required for pseudo-feature projection to calculate $\hat{\mu}_{c,f}$ and $\hat{\sigma}_{c,f}$. In the absence of statistical information, we fix the standard deviation to $\hat{\sigma}_{c,f} = \mathbf{1}$. This decision is based on the fact that the estimation of $\hat{\mu}_{c,f}$ already provides sufficient variance. Therefore, for the estimation of the mean component $\hat{\mu}_{c,f}$ we propose three heuristics:

1. **zeros**: clamping all $\hat{\mu}_{c,f}$ estimations to 0

$$\hat{\mu}_{c,f} = \mathbf{0}, \quad (6)$$

2. **random**: randomly sample $\hat{\mu}_{c,f}$ from a multivariate normal distribution

$$\hat{\mu}_{c,f} \sim \mathcal{N}(\mathbf{0}; \Sigma), \quad (7)$$

3. **original features**: estimate $\hat{\mu}_{c,f}$ with the original representation of the transforming sample and use them without modification

$$\hat{\mu}_{c,f} = \mathbf{f}(x_i; \theta). \quad (8)$$

Est. Method	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	Avg. Acc. \uparrow
zeros	73.3	39.9	32.9	35.2	25.2	25.4	24.9	19.2	18.3	19.1	17.1	30.0
random-1.0	73.3	39.9	32.9	35.2	25.2	25.4	24.8	19.2	18.3	19.0	17.2	30.0
random-3.0	73.3	47.0	37.9	38.0	29.0	31.3	30.0	22.9	21.9	22.5	22.5	34.2
random-5.0	73.3	52.5	42.7	44.4	33.6	34.5	33.8	25.2	25.9	26.3	26.4	38.0
random-10.0	73.3	59.2	50.8	52.7	43.5	43.4	40.2	31.4	31.8	31.8	31.7	44.5
random-15.0	73.3	61.7	53.5	54.2	45.6	46.1	42.0	35.8	36.1 ●	34.8 ●	34.4	47.0
random-20.0	73.3	62.5	55.5	54.8 ●	47.3	47.0	42.9	36.5	35.5	34.8 ●	34.4	47.7
random-30.0	73.3	62.9 ●	55.9	55.1 ●	48.8 ●	47.6	43.7 ●	36.8	36.0	34.8 ●	34.7 ●	48.1 ●
random-40.0	73.3	62.6	58.0 ●	54.8 ●	46.9	47.8 ●	44.9 ●	38.6 ●	36.5 ●	35.9 ●	34.9 ●	48.6 ●
random-50.0	73.3	64.1 ●	57.2	54.8 ●	47.6 ●	47.9 ●	42.0	37.6 ●	33.8	34.8 ●	34.4	48.0
random-75.0	73.3	62.1	57.9 ●	54.5	46.2	45.4	40.7	35.6	34.1	33.7	33.6	47.0
random-100.0	73.3	62.6	56.5	53.9	46.5	45.8	40.7	34.4	34.2	33.8	33.4	46.8
original features	73.3	64.3 ●	61.3 ●	60.8 ●	55.1 ●	53.7 ●	52.7 ●	46.8 ●	47.1 ●	46.3 ●	45.2 ●	55.2 ●

Table 1. Results for class prototype estimation when the corresponding class prototype is not available during training. The evaluation is performed on a CIL 50/10 setup with Slim-Resnet-18 only. **1st** ●, **2nd** ● and **3rd** ● best metrics are marked accordingly.

We evaluate the proposed feature estimation heuristics in an empirical experiment on a class-incremental learning scenario with no repetition. The results on CIFAR 50/10 trained on a Slim-Resnet-18 are listed in Table 1 (see Sec. 4 for more details). This scenario requires the estimation of class prototype components (e.g., mean, variance) at each incremental task and the estimation is essential for the classification. The *original features* estimation performed best, and this heuristic is the one used in all subsequent experiments.

In EFCIR scenarios, the repetition of classes within incremental tasks enhances the performance of pseudo-feature projection as it aligns individual FE representation spaces by eliminating the need for estimating class prototype components. During class repetition they can be directly calculated from the available task.

4. Experimental Setup

Most incremental learning methods expect a different set of classes with all dataset samples for each class available when learning its corresponding task. However, when class repetition is introduced, the complexity of potential scenarios increases significantly, and where sequence length and repetition frequency become additional variables. To address this, we propose an analysis into the effects of class repetition within a setting that shares many characteristics of traditional incremental learning but incorporates longer sequences with class repetition. Code for the proposed scenarios and methods is available¹.

Overall, the proposed experiments aim to analyze a) the performance of IL methods in scenarios without repetition (baseline), b) the performance of CIL with small incremental tasks and class repetition, and c) the resilience of the methods against bias deviations in repetition frequency.

Ideally, we expect the average accuracy of our proposed method to be on par with state-of-the-art methods

on (a) and to outperform them in (b) and (c). To validate this, method performance will be ranked based on average accuracy for all scenarios (a – c).

4.1. Compared Methods

We benchmark a total of 14 methods, which include two rehearsal-based approaches, five incremental learning methods, five state-of-the-art exemplar-free class-incremental learning (EFCIL) methods, and two variants of our proposed approach. The two rehearsal-based methods are excluded from the ranking and serve as an upper baseline (Joint [8]) and a reference point (Weight-Alignment (WA) [45]; $n = 2000$).

The five incremental learning methods consist of two baseline methods (Freezing (FZ) and Finetuning (FT) [27]), and three classic IL methods Elastic Weight Consolidation (EWC) [20], Memory Aware Synapses (MAS) [2] and Learning without Forgetting (LWF) [22]. These three methods were not originally proposed for CIL, thus, requiring the use of a task-ID at inference time. However, they are easily and commonly adaptable to task-agnostic settings. As such, we performed a grid search for their optimal hyperparameters based on the CIL 50/10 setting and used these for the repetition settings.

The five state-of-the-art, rehearsal-free, prototype-based methods comprise: Prototype Augmentation and Self-Supervision (PASS) [47], Class-Incremental Learning with Dual Supervision (IL2A) [46], Self-Sustaining Representation Expansion (SSRE) [48], Prototype Reminiscence and Augmented Asymmetric Knowledge Aggregation (PRAKA) [40] and Feature Translation for Exemplar-free Class Incremental Learning (FeTrIL) [33]. These methods were originally reported on the CIL CIFAR 50/10 setting. Therefore, since the proposed repetition scenarios are closely related to this setting, we use the hyperparameters proposed by the original authors.

Finally, we evaluate our proposed method with both ensemble growth heuristics (Horde_m and Horde_c). A de-

¹www.github.com/Tsebeb/cvww_cir_horde

tailed overview of the used hyperparameters is provided in the supplementary material (see Sec. C).

4.2. Model Architecture

All methods employ the same base feature extractor, a ResNet-18 [13] model that has been adjusted to the CIFAR input dimensions [46, 47]. For our approach, which utilizes an ensemble of feature extractors, we employ a slimmed-down variant of ResNet-18 for incremental tasks. This variant reduces the number of channels/filters for convolutions while preserving the network’s depth (see supplementary material Sec. B). With this reduced architecture, we construct an ensemble consisting of one full ResNet-18 and nine Slim-ResNet-18 models (making our budget $B=10$). This configuration results in a total number of parameters and computational requirements (see Table 2) that are roughly equivalent to those of knowledge distillation approaches (or importance weight estimation [2, 20, 23]).

4.3. Scenarios

Experiments are conducted on the CIFAR-100 dataset [21], employing data augmentation in line with other CIL methods [40, 46]. These augmentations consist of a 4-pixel zero padding of the input image and a random cropping to the original 32×32 size. Followed by a random horizontal flip, image brightness jitter and image normalization.

To evaluate the effects of repetition on CIL methods, we organize the experiments in three scenarios. First, a baseline is established by evaluating (a) all methods on an incremental learning scenario without repetition.

- (a) **CIL 50/10.** The classic task-agnostic class-incremental scenario consisting of an initial training session with 50 classes and followed by 10 incremental tasks, each containing five novel classes.

Second, we evaluate (b) performance on a modified CIL scenario where classes repeat in the task sequence. Specifically, the scenario is built by replacing the discrete incremental tasks with clear boundaries from CIL 50/10 with small (2,000 training samples per task) incremental tasks that can contain class repetition. Each class, old or new, has the same probability of being in an incremental task.

- (b) **EFCIR-U 50/100.** Similarly to the CIL 50/10 scenario, the initial training also covers 50 classes. An essential element of repetition is a mixture of new and already seen samples. Therefore, we only provide 50% of the available training data samples for the initial training. Following the initial task, the scenario consists of 99 small, incremental tasks, with a limit of 2,000 training samples each. Both the initial 50 and incremental 50 classes have a fixed probability of 15% of being discovered or repeated in an incremental task so that tasks do not contain too many classes on average. The number of samples per class in a task are balanced as in the CIL 50/10 scenario.

Model	# Parameters
ResNet-18	11.307.956
Slim ResNet-18	1.109.240
Knowledge Distillation	22.615.912
Ensemble (ours)	21.291.116

Table 2. Number of parameters for different architectures.

In the third scenario, the aim is to assess the IL method’s (c) resilience against biases in repetition frequency. To establish this bias during scenario creation we propose to draw the repetition probability of each class from a *Beta Distribution* [19]. An illustration of the repetition bias is provided in the supplementary material Sec. E.

- (c) **EFCIR-B 50/100.** To test the resiliency against repetition frequency, we sample individual class repetition probabilities $p \sim \text{Beta}(\alpha, \beta)$ with parameters $\alpha = 3.5$ and $\beta = 20.0$. This way, the expectation $\mathbb{E}[\text{Beta}(3.5, 20.0)] \approx 0.15$ is similar to the uniform EFCIR-U scenario, implying that on average the same number of classes are present in each task.

In scenarios with repeated classes, the optimal learning rate and number of epochs depend on various factors (*e.g.* method, number of training samples, length of incremental sequence) and are highly influential. To address this, we split 10% of the available training data as a validation set. For all methods, we apply early stopping [5, 34] using this validation data for the classes present in the task. We monitor the validation loss (including regularization and auxiliary losses of the method) and allow for a patience period of 5 epochs. If no improvement is observed, we perform a learning rate decay step. Each decay step reduces the learning rate by a factor of 0.1, the model weights are reset to the best checkpoint before patience, and we do not perform more than 2 decay steps.

Evaluation. All scenarios are ranked by the average accuracy [8, 27, 33, 36, 47] achieved over the complete task sequence. Average accuracy is calculated by evaluating the model on the CIFAR test set based on the classes that have been seen up to each task. Complementary to the average accuracy, average *forgetting* [6, 8, 27] is also reported, which measures the drop in accuracy over the task sequence. Experimental results are averaged over 5 seeds.

5. Results

The summarized results of the experiments are listed in Table 3. Detailed plots and tables of the accuracy progression for all methods in each proposed scenario are provided in the supplementary material (Sec. G).

Scenario (a) CIL 50/10. The conducted baseline experiment confirms the reported results from other works [8, 40, 46, 47]. We observe a significant performance gain of approximately 10-15% in average accuracy over the task

	(a) CIL 50/10		(b) EFCIR-U 50/100		(c) EFCIR-B 50/100	
Method	Avg. $A \uparrow$	Avg. $f \downarrow$	Avg. $A \uparrow$	Avg. $f \downarrow$	Avg. $A \uparrow$	Avg. $f \downarrow$
Joint	73.9	-	69.8	-	68.9	-
WA [45]	42.7 ± 2.3	33.2 ± 1.4	50.4 ± 0.2	16.7 ± 2.4	49.2 ± 0.7	18.0 ± 1.6
FT	14.2 ± 1.0	57.8 ± 1.2	36.2 ± 2.1	25.6 ± 2.7	34.2 ± 2.0	29.0 ± 2.6
FZ	52.6 ± 1.4	19.7 ± 0.9	40.2 ± 3.9	20.0 ± 1.6	41.7 ± 3.1	22.5 ± 1.8
EWC [20]	45.9 ± 2.9	25.7 ± 1.4	47.7 ± 3.2	13.5 ± 1.5 \bullet	45.5 ± 3.2	17.8 ± 1.8 \bullet
MAS [2]	45.9 ± 2.9	25.8 ± 1.4	49.3 ± 2.6 \bullet	12.0 ± 1.8 \bullet	47.2 ± 2.3 \bullet	16.1 ± 2.1 \bullet
LwF [22]	47.9 ± 1.8	24.1 ± 0.8	45.7 ± 1.9	15.9 ± 2.8 \bullet	43.5 ± 0.8	19.8 ± 4.1
PASS [47]	62.1 ± 1.9	14.1 ± 0.4	30.2 ± 2.0	35.3 ± 2.1	30.6 ± 1.4	38.3 ± 1.6
PRAKA [40]	63.1 ± 2.5 \bullet	11.8 ± 2.2 \bullet	43.1 ± 2.1	22.3 ± 2.3	42.7 ± 3.2	25.6 ± 1.8
IL2A [46]	54.2 ± 1.4	19.1 ± 1.3	26.3 ± 3.0	32.2 ± 2.9	27.2 ± 2.5	37.2 ± 1.7
SSRE [48]	53.0 ± 2.7	13.0 ± 0.8 \bullet	29.2 ± 3.5	25.4 ± 2.1	26.5 ± 2.2	26.4 ± 2.1
FeTrIL [33]	61.4 ± 0.4	13.6 ± 0.8 \bullet	46.5 ± 0.7	22.9 ± 0.7	46.9 ± 0.9	23.8 ± 1.2
<i>Horde_m</i>	62.9 ± 1.2 \bullet	15.2 ± 0.7	54.4 ± 0.7 \bullet	16.4 ± 1.5	54.3 ± 0.4 \bullet	17.7 ± 1.0 \bullet
<i>Horde_c</i>	62.9 ± 1.2 \bullet	15.3 ± 0.6	53.4 ± 0.7 \bullet	17.6 ± 1.6	53.1 ± 0.4 \bullet	18.5 ± 1.1

Table 3. Average Accuracy (Avg. A) and average Forgetting (Avg. f) for all 3 proposed scenarios. The listed results are averaged over 5 seeds (except incremental Joint). The 3 best results are marked with a gold \bullet , silver \bullet and bronze \bullet medal respectively.

sequence when comparing the state-of-the-art rehearsal-free (EFCIL) methods with EWC, MAS and LwF. While our proposed method is particularly designed towards repetition scenarios, where the estimation of class prototype components is not always required, it remains competitive in disjoint, no-repetition scenarios as well, showing comparable performance to the best EFCIL methods [33, 40, 46, 47].

Scenario (b) EFCIR-U. Introducing class repetition in small incremental tasks into the scenario leads to significant performance differences. Weight-regularization approaches and vanilla finetuning typically underperform compared to knowledge distillation or class prototype-based approaches in EFCIL [40]. However, in this scenario with repetition, we observe greatly improved performance for FT, EWC and MAS. The results for these methods surpass even the results from the CIL 50/10 scenario by leveraging data repetition effectively (see Fig. 3). In contrast, EFCIL methods (PASS, IL2A, SSRE, PRAKA) that rely on both class prototype rehearsal and knowledge distillation show a performance degradation under repetition. This decline is not observed in methods that use either knowledge distillation (LwF) or class prototype rehearsal with frozen feature extractors (FeTrIL, Ours).

We hypothesize that the estimation of class prototypes with incomplete class data distribution in the former methods leads to a suboptimal feature embedding space, which is then propagated through the incremental task sequence via knowledge distillation. Frozen feature extractors, on the other hand, avoid this issue since their representations remain fixed after the initial training, preventing catastrophic drift in the embedding space during the task sequence. This raises the question whether the assumption that the complete training data distribution of an individual class – as in traditional class-incremental learning – is a realistic assumption for continual learning scenarios.

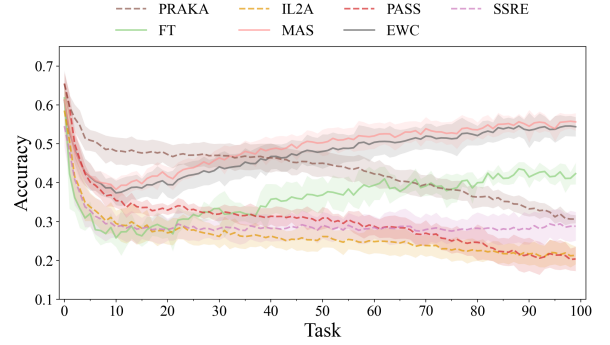
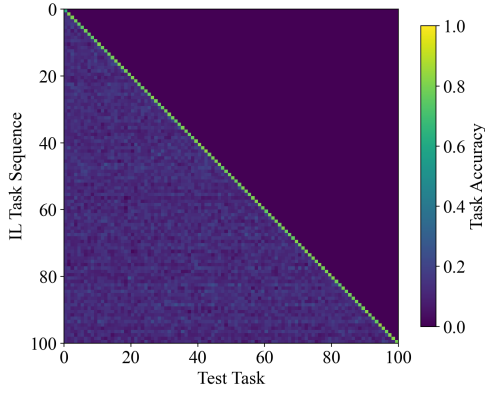


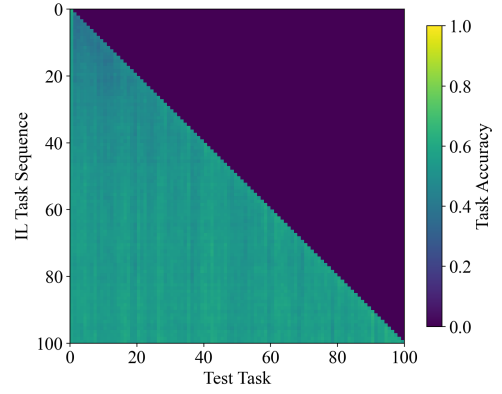
Figure 3. Accuracy curves for scenario (b) with equiprobable repetition frequency. Weight-regularized methods (*solid*) benefit directly from short tasks with class repetition, while prototype-based approaches (*dashed*) degrade in accuracy as the sequence advances.

Our ensemble-based approach (Horde), with both ensemble growth heuristics, establishes a new state-of-the-art for the repetition settings. Notably, when comparing our method with the closely related FeTrIL approach, we observe a performance increase under repetition. This suggests that our approach could extend the feature space of the base feature extractor by incorporating class combinations from smaller feature extractors. Over time, repetition aligns these representations, enabling the model to learn a unified classification head on a more diverse representation space provided by the ensemble.

The strong performance gains for weight-regularized approaches are only observed when the cross-entropy loss during training is limited to the classes that are present within the current task. Practically, this is achieved by freezing all weights associated with classes outside of the current task [27]. Figures 4 and 5 illustrate the consequences of backpropagating the loss through all weights of the classification head. In this case, regardless of



(a) Cross-entropy loss gradient applied to all class weights in the classification head. Significant task recency bias is visible (the diagonal is significantly higher, with sharp drops after each task).



(b) Cross-entropy loss gradient applied to only current task class weights in the classification head. Much of the task recency bias is alleviated by freezing the classifier weights for unavailable classes.

Figure 4. Depiction of the task accuracy progression of MAS over the scenario (b) sequence (averaged over 5 seeds). Accuracy is evaluated on the test set for the classes represented in the corresponding incremental training data within a task. Note, that for repetition there is always a certain overlap within tasks.

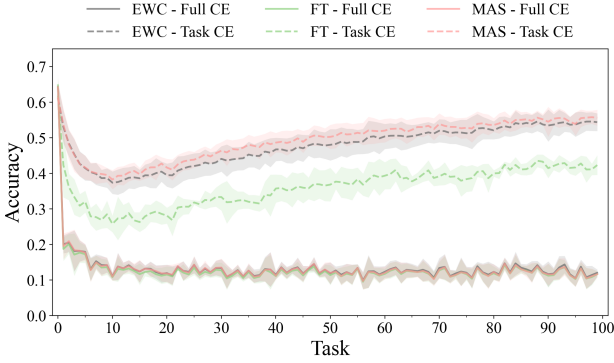


Figure 5. Accuracy results for finetuning and weight-regularization based methods. Solid lines indicate the backpropagation of the cross-entropy loss over all classes leading to catastrophic class recency bias. Dashed lines indicate the freezing of the weights related to the output of non-current classes.

whether weight-regularization is applied to the feature extractor, a strong class recency bias emerges in the classification head. As a result, the accuracy of all three methods collapses, with the model essentially forgetting classes proportionally to how long they were seen last (see Fig. 4). However, when the weights for unavailable classes in the classification head are frozen, a significant performance improvement is observed, as the model retains its ability to distinguish across earlier tasks without being overly biased towards the most recent ones.

Scenario (c) EFCIR-B. The bias in repetition frequency appears to have only a minor effect on the average accuracy of the approaches. All tested methods achieve similar results or experience only a slight drop of up to $\sim 2\%$ in average accuracy. This suggests that repetition frequency bias is a relatively minor challenge in the EFCIR-B 50/100 scenario. However, it is important to note that this setting

only evaluates adjustments in repetition frequency while the sample distribution within a training task is kept balanced. Therefore, further investigation is needed to assess whether an imbalanced training data distribution in conjunction with biased repetition frequency would increase the difficulty. We leave this exploration to future work.

6. Conclusion

In this work, we conducted an exploratory evaluation of CIL methods in exemplar-free class-incremental learning with repetition scenarios and investigated their resiliency to biases in the repetition frequency of classes.

In the evaluated repetition scenarios, EFCIL methods that rely on class prototypes (PASS, PRAKA, IL2A, SSRE) severely underperform and are unable to benefit from the repetition of classes. Notably, weight-regularization-based approaches perform exceptionally well in repetition scenarios provided that training with cross-entropy is restricted to the classes present in each task, thereby mitigating the risk of class-recency bias in the classification head. The results from the repetition frequency bias from a beta distribution show only minimal performance differences, with either no effect on average accuracy or a slight drop of up to 2%. Thus, a bias in repetition frequency alone without a biased sample distribution within a training task is insufficient for significant classification bias.

Furthermore, we introduce a novel ensemble learning technique that takes advantage of class repetition. This method combines a dynamic set of independent feature extractors, which are aligned through a unified head in a process we call pseudo-feature projection. The proposed method demonstrates competitive performance in traditional no-repetition settings and establishes a new state-of-the-art for scenarios with repetition.

Acknowledgements

Marc Masana acknowledges the support by the “University SAL Labs” initiative of Silicon Austria Labs (SAL).

References

- [1] 5th CLVISION CVPR workshop challenge, 2023. [2](#)
- [2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018. [2](#), [5](#), [6](#), [7](#), [12](#)
- [3] Eden Belouadah, Adrian Popescu, Umang Aggarwal, and Léo Saci. Active class incremental learning for imbalanced datasets. In *European Conference on Computer Vision*, pages 146–162. Springer, 2020. [1](#)
- [4] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021. [1](#), [3](#)
- [5] Christopher M. Bishop. Regularization and complexity control in feed-forward networks. International Conference on Artificial Neural Networks ICANN’95., 1995. [6](#)
- [6] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018. [6](#)
- [7] Andrea Cossu, Gabriele Graffieti, Lorenzo Pellegrini, Davide Maltoni, Davide Bacciu, Antonio Carta, and Vincenzo Lomonaco. Is class-incremental enough for continual learning? *Frontiers in Artificial Intelligence*, 5:829842, 2022. [1](#), [2](#)
- [8] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. [1](#), [2](#), [4](#), [5](#), [6](#)
- [9] Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. Don’t forget, there is more than forgetting: new metrics for continual learning. *arXiv preprint arXiv:1810.13166*, 2018. [1](#)
- [10] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. [1](#), [2](#)
- [11] Mudassir A Ganaie, Minghui Hu, AK Malik, M Tanveer, and PN Suganthan. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151, 2022. [2](#)
- [12] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. [1](#), [2](#)
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [6](#), [11](#)
- [14] Lu Yu He Li and Wu He. The impact of gdpr on global technology development. *Journal of Global Information Technology Management*, 22(1):1–6, 2019. [1](#)
- [15] Hamed Hemati, Andrea Cossu, Antonio Carta, Julio Hurtado, Lorenzo Pellegrini, Davide Bacciu, Vincenzo Lomonaco, and Damian Borth. Class-incremental learning with repetition. *arXiv preprint arXiv:2301.11396*, 2023. [2](#)
- [16] Hamed Hemati, Lorenzo Pellegrini, Xiaotian Duan, Zixuan Zhao, Fangfang Xia, Marc Masana, Benedikt Tscheschner, Eduardo Veas, Yuxiang Zheng, Shiji Zhao, Shao-Yuan Li, Sheng-Jun Huang, Vincenzo Lomonaco, and Gido M. van de Ven. Continual learning in the presence of repetition. *Neural Networks*, 183:106920, 2025. [2](#)
- [17] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. [12](#)
- [18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. [2](#)
- [19] Norman L Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous univariate distributions, volume 2*. John Wiley & sons, 1995. [6](#)
- [20] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. [2](#), [5](#), [6](#), [7](#), [12](#)
- [21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. [6](#)
- [22] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. [2](#), [5](#), [7](#), [12](#)
- [23] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 2262–2268. IEEE, 2018. [6](#)
- [24] David Lopez-Paz and Marc’Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. [1](#)
- [25] Marc Masana, Idoia Ruiz, Joan Serrat, Joost van de Weijer, and Antonio M Lopez. Metric learning for novelty and anomaly detection. In *British Machine Vision Conference (BMVC)*, 2018. [3](#)
- [26] Marc Masana, Tinne Tuytelaars, and Joost van de Weijer. Ternary feature masks: continual learning without any forgetting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2021. [2](#)
- [27] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2023. [1](#), [2](#), [5](#), [6](#), [7](#)
- [28] Benjamin Maschler, Thi Thu Huong Pham, and Michael Weyrich. Regularization-based continual learning for anomaly detection in discrete manufacturing. *Procedia CIRP*, 104:452–457, 2021. 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0. [1](#)

- [29] Martial Mermillod, Aurélie Bugaiska, and Patrick BONIN. The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in Psychology*, 4, 2013. 1, 2
- [30] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019. 1
- [31] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113: 54–71, 2019. 1
- [32] Grégoire Petit, Adrian Popescu, Eden Belouadah, David Picard, and Bertrand Delezoide. Plastil: Plastic and stable memory-free class-incremental learning. In *Second Conference on Lifelong Learning Agents (CoLLAs)*, 2023. 2
- [33] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3911–3920, 2023. 2, 3, 4, 5, 6, 7, 12
- [34] Simon J.D. Prince. *Understanding Deep Learning*. MIT Press, 2023. 3, 6
- [35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1
- [36] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2, 6
- [37] Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 3
- [38] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2
- [39] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR, 2018. 1
- [40] Wuxuan Shi and Mang Ye. Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1772–1781, 2023. 2, 5, 6, 7, 12
- [41] Guido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1, 2
- [42] Guido M Van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1): 4069, 2020. 1
- [43] Eli Verwimp, Rahaf Aljundi, Shai Ben-David, Matthias Bethge, Andrea Cossu, Alexander Gepperth, Tyler L Hayes, Eyke Hüllermeier, Christopher Kanan, Dhireesha Kudithipudi, et al. Continual learning: Applications and the road forward. *arXiv preprint arXiv:2311.11908*, 2023. 1
- [44] Dejie Yang, Minghang Zheng, Weishuai Wang, Sizhe Li, and Yang Liu. Recent advances in class-incremental learning. In *Image and Graphics*, pages 212–224, Cham, 2023. Springer Nature Switzerland. 2
- [45] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13208–13217, 2020. 5, 7, 12
- [46] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34: 14306–14318, 2021. 2, 5, 6, 7, 12
- [47] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5871–5880, 2021. 2, 3, 5, 6, 7, 11, 12
- [48] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022. 2, 5, 7, 12

Incremental Learning with Repetition via Pseudo-Feature Projection

Supplementary Material

A. Pseudo Code

The algorithm for the proposed Horde method can be separated into two parts: (1st) the training of an individual feature extractor (FE), which is listed in Algorithm 1 and (2nd) the overall assembly of the ensemble and training of the unified head through pseudo-feature projection (see Algorithm 2). The training of a feature extractor (1st part) can be freely adjusted (loss, network architecture) as long as a frozen feature extractor that can produce an embedding is the result.

Algorithm 1 FE Training

```

1: Initialize CE and ML Head
2: Initialize new FE (or transfer learned weights)
3: for training epoch do
4:   for  $X; Y$  in Dataloader do
5:      $X; Y \leftarrow \text{SelfSupervision}(X; Y)$ 
6:     Extract  $\hat{X} \leftarrow \text{FE}(X)$ 
7:     Predict  $\hat{Y} \leftarrow \text{Head}_{\text{CE}}(\hat{X})$ 
8:     Project  $A \leftarrow \text{Head}_{\text{ML}}(\hat{X})$ 
9:     Calculate  $\mathcal{L}_{\text{CE}}$  (from  $Y$  and  $\hat{Y}$ )
10:    Calculate  $\mathcal{L}_{\text{ML}}$  (with Hard Neg. Pairs on  $A$ )
11:    Backprop  $\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{ML}}$ 
12:   end for
13: end for
14: Remove CE and ML head
15: Freeze FE

```

Algorithm 2 IL through pseudo-feature projection

```

1: for task do
2:   if Growth Condition ( $\text{Horde}_m$  or  $\text{Horde}_c$ ) then
3:     Train FE (Algorithm 1)
4:     Add / Replace FE in ensemble
5:   end if
6:   Calculate  $\mu_c$  and  $\sigma_c$  for all current classes  $c$ 
7:   for training epoch do ▷ Only Unified head
8:     for Batch do
9:       Calculate  $\mathcal{L}_{\text{CE}}$ 
10:      Generate  $\hat{F}_c$  from current Batch
11:      Calculate  $\mathcal{L}_{\text{CE};P}$  for  $\hat{F}_c$ 
12:      Backprop  $\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{CE};P}$ 
13:    end for
14:   end for
15: end for

```

B. Details about the Model Architecture

The individual layers of a ResNet-18 are listed in Table 4 and a structural overview is depicted in Figure 6. There is no difference in the depth of the network or the type

ResNet-18		
$C_b = 20$ for SlimResNet18 and $C_b = 64$ for ResNet-18		
Layer	Stride	Dimension
Conv 3×3	1	$C_b \times 32 \times 32$
BatchNorm	-	$C_b \times 32 \times 32$
ReLU	-	$C_b \times 32 \times 32$
BasicBlock $C_{in} = C_b, C_{out} = C_b$	1	$C_b \times 32 \times 32$
BasicBlock $C_{in} = C_b, C_{out} = 2 \cdot C_b$	2	$2 \times C_b \times 16 \times 16$
BasicBlock $C_{in} = 2 \cdot C_b, C_{out} = 3 \cdot C_b$	2	$3 \times C_b \times 8 \times 8$
BasicBlock $C_{in} = 3 \cdot C_b, C_{out} = 4 \cdot C_b$	2	$4 \times C_b \times 4 \times 4$
AvgPool 4×4	1	$4 \times C_b \times 1 \times 1$
Linear (Classification Head)	-	#classes

Table 4. The network structure is identical for both ResNet-18 and its SlimResNet-18 variant, besides a reduction in the number of base channels C_b .

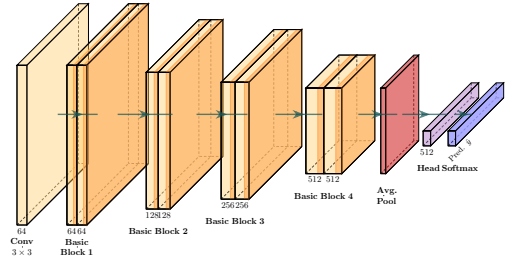


Figure 6. Visualization of the structure of a ResNet-18 [13].

of layers between the ResNet-18 and the Slim-ResNet-18. The only difference is the number of base filters C_b for the convolutions in the *Basic Blocks*. The Slim-ResNet-18 uses $C_b = 20$ while the full ResNet-18 uses $C_b = 64$. This influences the number of channels for the following operations so that a compression to approximately a tenth of the original size can be achieved.

C. Method Hyperparameters

The hyperparameters for all compared methods are listed in Table 5. For EWC, MAS and LWF, we perform a grid-search over their main hyperparameters on the CIL 50/10 scenario, and the one achieving the highest average accuracy are fixed for the repetition tasks. The remaining hyperparameters are the ones recommended by their original authors for the corresponding scenario.

D. Feature Extractor Training Components

Table 6 provides an overview of the effects of each component in the Feature Extractor and its effect on the average accuracy in the CIL scenario (a). The results have been averaged over 5 seeds. Both the self-supervision from PASS [47] as well as the training with the metric learning head are beneficial based on the overall average accuracy. The metric learning head alone without a cross-

Method	Hyperparameter
FT	-
FZ	freeze after 1st task
Joint	-
WA [45]	2000 exemplars, $\tau = 2$, <i>patience</i> = 10
EWC [20]	$\lambda = 40000$, $\alpha = 0.1$
MAS [2]	$\lambda = 10$, $\alpha = 0.1$
LwF [22]	$\lambda = 30$, $\tau = 2$
PASS [47]	$\tau_{CE} = 0.1$, $\tau_{KD} = 2$, $\lambda_{kd} = 10.0$, $\lambda_{aug} = 10.0$
IL2A [46]	$\tau_{CE} = 0.1$, $\lambda_{KD} = 10.0$, $\lambda_{seman} = 10.0$, $\#mixups = 4$
PRAKA [40]	$\tau_{CE} = 0.1$, $\lambda_{aug} = 15.0$, $\lambda_{KD} = 15.0$
SSRE [48]	$\tau_{CE} = 0.1$, $\lambda_{aug} = 10.0$, $\lambda_{KD} = 1.0$
FeTrIL [33]	AugMix [17] pre-train, fc head, 1-cosine translation
Horde (ours)	original features estimation, CE & ML Head, self-supervision, 1 Resnet18, 9 Slim Resnet18s

Table 5. Overview of approach-specific hyperparameters

CE-Head	ML-Head	Self-Supervision [47]	Avg. Acc \uparrow
✓	✗	✗	56.91 ± 0.88
✗	✓	✗	7.72 ± 0.85
✓	✓	✗	58.68 ± 0.79 ●
✓	✗	✓	60.62 ± 1.21 ●
✗	✓	✓	9.76 ± 1.15
✓	✓	✓	63.09 ± 1.19 ●

Table 6. Ablation study results on different variations of FE training. **1st** ●, **2nd** ● and **3rd** ● best metrics are marked accordingly.

entropy head is however insufficient for the training of a feature extractor.

E. Scenario Visualization

In the proposed experiments we differentiate between a fairly balanced repetition scenario and a biased scenario. The difference between the two repetition frequencies is visualized in Fig. 7 and Fig. 8. On average both scenarios have 15 classes in each incremental task.

F. Longer Task Sequence

The results from scenario (b) indicate a strong accuracy recovery/trend for weight regularization techniques. We further evaluate with even longer task sequences where the number of incremental tasks is increased from 99 to 149. The accuracy on later tasks is very strong on weight-regularization techniques as the overall accuracy trend continues. However, it is important to note that, already in the 100 task scenario, all available training data is used in the task sequence at least once, thus further tasks can only repeat samples and no longer provide any new/incremental training data. Although EWC and MAS both achieve a significant higher final accuracy in the longer task sequence, they are still slightly worse in terms of average accuracy across the whole sequence,

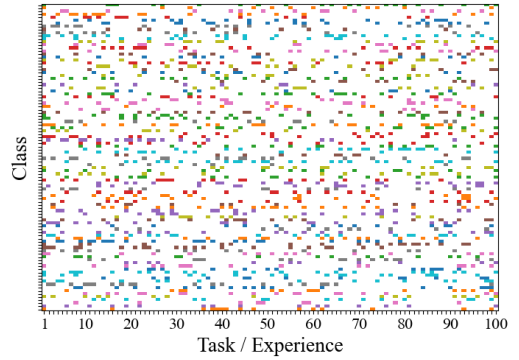


Figure 7. Class distribution visualization of scenario (b), with uniform class occurrence frequency. Each colored block indicates that the class is sampled in the corresponding task.

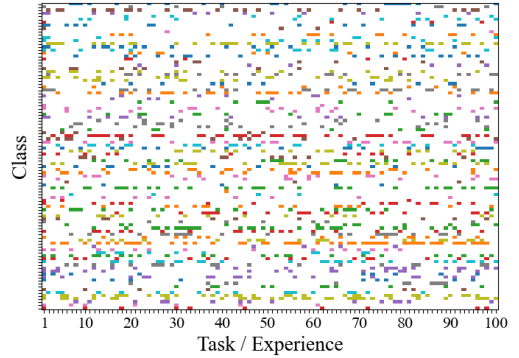


Figure 8. Class distribution visualization of scenario (c), with biased (beta) class occurrence frequency. Each colored block indicates that the class is sampled in the corresponding task.

since they are less stable in the initial tasks of the sequence. The compared average accuracies for the 100 and 150 task scenarios, as well as final test accuracy after the task sequence, are listed in Table 7. Furthermore, the accuracy progression is visualized in Figure 9.

Supplementary Material

Method	Avg. A_{100}	Avg. A_{150}	final A_{100}	final A_{150}
FT	36.2 ± 2.1	39.3 ± 2.1	42.3 ± 2.7	46.9 ± 1.1
EWC	47.7 ± 3.2	51.4 ± 0.9	54.4 ± 2.5 ●	57.2 ± 0.7 ●
MAS	49.3 ± 2.6 ●	52.5 ± 0.8 ●	55.6 ± 2.2 ●	59.0 ± 0.3 ●
$Horde_c$	54.4 ± 0.7 ●	53.2 ± 1.6 ●	55.1 ± 0.7 ●	54.4 ± 0.4 ●
$Horde_m$	53.4 ± 0.7 ●	53.8 ± 0.9 ●	54.0 ± 1.1	53.4 ± 1.9

Table 7. Comparison between unbiased repetition scenarios of 100 and 150 tasks. While our proposed method is more stable, especially in the initial phases of training. The trend of weight regularization methods continues and the final accuracy continues to increase.

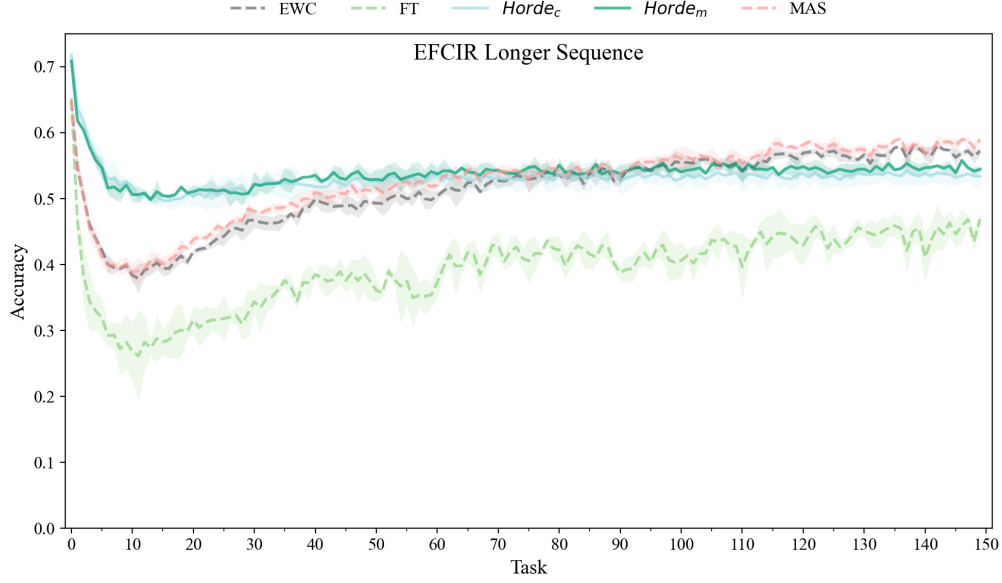


Figure 9. Average Accuracy over an even longer repetition scenario to analyse the trends between different methods. The performance increase of weight-regularization techniques continue

G. Scenario Results

The following figures visualize the detailed Average Accuracy development over the incremental task sequence. For each method the mean and one standard deviation have been plotted. The results for the class-incremental scenario (a) are listed in Table 8 and visualized in Figure 10. The unbiased repetition results for scenario (b) can be found in Table 9 and Figure 11. The results of the biased class-repetition scenario (c) are shown in Table 10 and Figure 12.

Supplementary Material

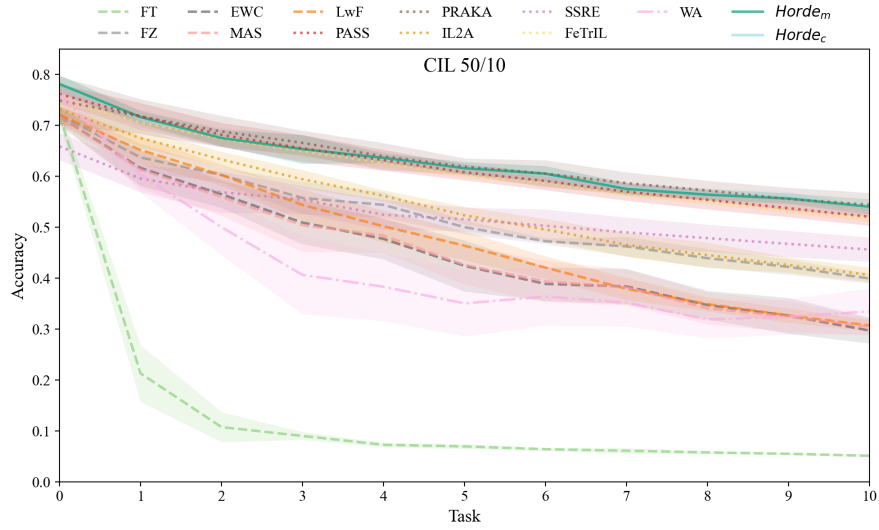


Figure 10. Accuracy development over the task sequence of scenario (a).

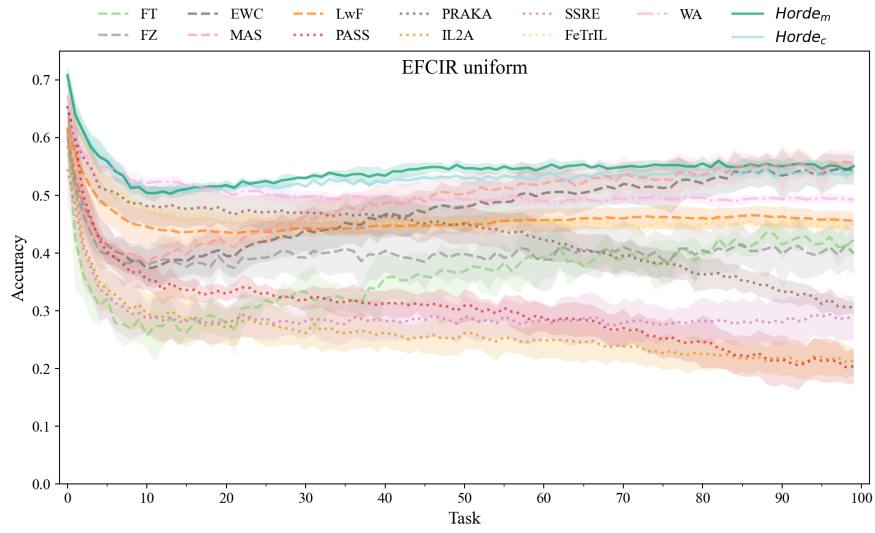


Figure 11. Accuracy development over the task sequence of scenario (b).

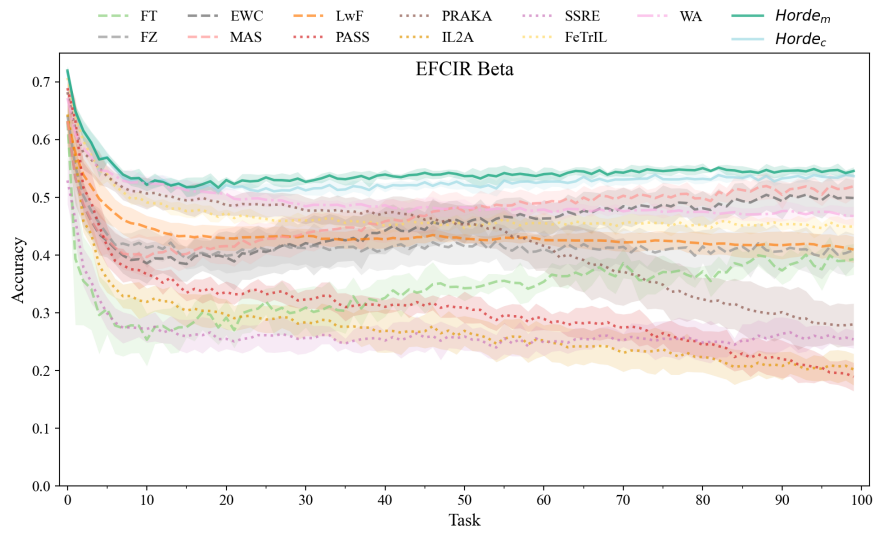


Figure 12. Accuracy development over the task sequence of scenario (c).

Supplementary Material

Method	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	Avg. $A \uparrow$	Avg. $f \downarrow$
WA	76.0 \pm 1.5	60.4 \pm 3.5	50.0 \pm 5.4	40.7 \pm 7.7	38.3 \pm 6.7	35.1 \pm 6.5	36.4 \pm 5.7	35.2 \pm 4.7	31.9 \pm 3.7	32.4 \pm 3.5	33.5 \pm 4.5	42.7 \pm 2.3	33.3 \pm 1.5
FT	72.1 \pm 1.7	21.3 \pm 5.5	10.8 \pm 2.9	9.0 \pm 0.8	7.3 \pm 0.4	7.0 \pm 0.5	6.4 \pm 0.3	6.1 \pm 0.5	5.8 \pm 0.3	5.5 \pm 0.2	5.1 \pm 0.3	14.2 \pm 1.0	57.9 \pm 1.2
FZ	72.1 \pm 1.7	63.7 \pm 2.7	60.3 \pm 1.4	55.8 \pm 3.2	54.4 \pm 2.5	50.0 \pm 1.4	47.3 \pm 0.5	46.2 \pm 1.9	43.9 \pm 1.8	42.2 \pm 1.5	39.9 \pm 0.9	52.4 \pm 1.4	19.7 \pm 0.9
EWC	71.7 \pm 1.6	61.6 \pm 3.5	56.5 \pm 2.6	50.9 \pm 4.2	47.7 \pm 4.0	42.4 \pm 5.0	38.9 \pm 3.3	38.4 \pm 3.5	34.7 \pm 2.8	32.6 \pm 3.5	29.7 \pm 2.5	45.9 \pm 3.0	25.8 \pm 1.5
MAS	71.7 \pm 1.6	61.4 \pm 3.6	55.9 \pm 3.6	50.4 \pm 5.3	48.4 \pm 3.3	42.6 \pm 3.8	39.4 \pm 4.1	38.2 \pm 3.6	34.1 \pm 3.2	32.5 \pm 3.0	30.5 \pm 1.7	45.9 \pm 2.9	25.8 \pm 1.4
LwF	72.1 \pm 1.7	65.2 \pm 2.7	60.3 \pm 2.0	54.4 \pm 3.1	50.2 \pm 3.1	46.4 \pm 2.9	42.1 \pm 2.2	37.9 \pm 2.2	34.9 \pm 1.3	32.7 \pm 1.4	30.7 \pm 1.1	47.9 \pm 1.8	24.2 \pm 0.8
PASS	76.2 \pm 2.0 ●	71.8 \pm 2.6 ●	68.2 \pm 2.3 ●	65.5 \pm 2.6 ●	63.2 \pm 2.0	60.8 \pm 1.5	59.1 \pm 1.8	57.0 \pm 1.7	55.4 \pm 1.9	53.8 \pm 1.7	52.1 \pm 1.7	62.1 \pm 1.9	14.1 \pm 0.5
PRAKA	74.9 \pm 4.7	71.8 \pm 3.4 ●	68.8 \pm 3.1 ●	66.6 \pm 2.4 ●	63.9 \pm 2.6 ●	61.9 \pm 1.6 ●	60.5 \pm 2.6 ●	58.7 \pm 2.1 ●	57.2 \pm 1.9 ●	55.6 \pm 2.2 ●	54.4 \pm 2.4 ●	63.1 \pm 2.6 ●	11.8 \pm 2.3 ●
IL2A	73.2 \pm 0.8	67.5 \pm 1.8	63.3 \pm 1.5	59.4 \pm 1.8	56.2 \pm 1.1	52.4 \pm 1.7	49.6 \pm 2.1	46.6 \pm 2.3	44.6 \pm 2.4	42.6 \pm 1.5	40.7 \pm 1.4	54.2 \pm 1.4	19.0 \pm 1.3
SSRE	65.9 \pm 2.6	59.6 \pm 2.6	56.9 \pm 4.0	55.4 \pm 2.7	52.5 \pm 2.9	51.6 \pm 2.3	50.3 \pm 3.2	49.0 \pm 2.9	47.9 \pm 2.7	46.7 \pm 2.6	45.6 \pm 2.4	52.9 \pm 2.7	13.0 \pm 0.8 ●
FeTrIL	75.0 \pm 1.2	70.6 \pm 0.9	67.4 \pm 0.9	64.9 \pm 1.1	62.6 \pm 0.6	60.2 \pm 0.5	58.7 \pm 0.5	56.5 \pm 0.5	55.0 \pm 0.5	53.2 \pm 0.4	51.6 \pm 0.6	61.4 \pm 0.4	13.6 \pm 0.8 ●
$Horde_m$	78.1 \pm 1.7 ●	71.6 \pm 1.3 ●	67.5 \pm 1.6	65.3 \pm 2.8 ●	63.6 \pm 1.0 ●	61.6 \pm 1.2 ●	60.5 \pm 1.5 ●	57.5 \pm 1.2 ●	56.4 \pm 1.1 ●	55.7 \pm 1.0 ●	54.0 \pm 1.5 ●	62.9 \pm 1.2 ●	15.2 \pm 0.7
$Horde_c$	78.2 \pm 1.7 ●	71.2 \pm 1.4	67.7 \pm 1.9 ●	65.2 \pm 2.8	63.4 \pm 0.8 ●	61.8 \pm 1.2 ●	60.2 \pm 1.9 ●	57.9 \pm 1.0 ●	56.6 \pm 1.1 ●	55.9 \pm 1.1 ●	53.8 \pm 0.4 ●	62.9 \pm 1.2 ●	15.3 \pm 0.7

Table 8. Results for the baseline CIL 50/10 scenario (a). **1st** ●, **2nd** ● and **3rd** ● best metrics are marked accordingly.

Method	A_0	A_{10}	A_{20}	A_{40}	A_{60}	A_{80}	A_{99}	Avg. $A \uparrow$	Avg. $f \downarrow$
WA	67.1 \pm 2.4	52.2 \pm 0.8	50.6 \pm 1.1	49.9 \pm 0.7	49.1 \pm 0.8	49.6 \pm 0.9	49.2 \pm 0.8	50.4 \pm 0.2	16.7 \pm 2.4
FT	61.8 \pm 4.3	25.8 \pm 2.3	28.1 \pm 2.2	35.7 \pm 3.9	39.3 \pm 3.8	40.0 \pm 0.9	42.3 \pm 2.7	36.2 \pm 2.1	25.6 \pm 2.7
FZ	60.1 \pm 4.8	37.9 \pm 3.7	37.9 \pm 3.4	40.4 \pm 3.5	38.9 \pm 5.5	40.5 \pm 3.6	40.0 \pm 3.6	40.2 \pm 4.0	20.0 \pm 1.6
EWC	61.1 \pm 4.1	37.4 \pm 3.1	39.5 \pm 3.3	46.7 \pm 3.3	50.4 \pm 3.5	52.1 \pm 2.5	54.4 \pm 2.5 ●	47.7 \pm 3.2	13.5 \pm 1.5 ●
MAS	61.3 \pm 4.1	38.2 \pm 2.6	41.6 \pm 2.5	48.6 \pm 2.7 ●	52.0 \pm 3.3 ●	53.6 \pm 1.9 ●	55.6 \pm 2.2 ●	49.3 \pm 2.6 ●	12.0 \pm 1.8 ●
LwF	61.6 \pm 4.2	44.5 \pm 3.0	43.6 \pm 2.2	44.7 \pm 1.5	45.7 \pm 2.0	46.1 \pm 1.9	45.6 \pm 1.7	45.7 \pm 1.9	15.9 \pm 2.8 ●
PASS	65.5 \pm 2.7	35.4 \pm 1.8	32.9 \pm 1.2	31.4 \pm 2.1	28.8 \pm 2.2	24.6 \pm 4.6	20.4 \pm 3.1	30.2 \pm 1.9	35.3 \pm 2.2
PRAKA	65.4 \pm 3.3	48.1 \pm 3.5 ●	47.2 \pm 2.9 ●	46.4 \pm 3.7	42.2 \pm 3.0	36.3 \pm 2.7	30.6 \pm 1.0	43.1 \pm 2.1	22.3 \pm 2.4
IL2A	58.5 \pm 5.8	29.5 \pm 3.3	27.1 \pm 3.0	26.3 \pm 2.3	24.8 \pm 3.0	22.5 \pm 2.6	21.3 \pm 2.3	26.3 \pm 3.0	32.2 \pm 2.9
SSRE	54.5 \pm 5.3	28.7 \pm 2.7	27.9 \pm 2.8	28.2 \pm 3.4	28.5 \pm 2.8	28.3 \pm 4.1	28.8 \pm 3.7	29.2 \pm 3.5	25.4 \pm 2.1
FeTrIL	69.3 \pm 1.1 ●	47.5 \pm 0.9	46.2 \pm 1.0	45.2 \pm 1.4	45.4 \pm 1.1	45.0 \pm 1.5	45.2 \pm 0.3	46.5 \pm 0.7	22.9 \pm 0.7
$Horde_m$	70.8 \pm 1.7 ●	50.4 \pm 1.1 ●	51.7 \pm 1.0 ●	53.4 \pm 1.1 ●	54.8 \pm 1.1 ●	55.5 \pm 1.3 ●	55.1 \pm 0.7 ●	54.4 \pm 0.7 ●	16.4 \pm 1.5
$Horde_c$	70.9 \pm 1.9 ●	50.9 \pm 1.0 ●	51.7 \pm 0.9 ●	52.1 \pm 0.7 ●	53.6 \pm 1.2 ●	53.4 \pm 1.4 ●	54.0 \pm 1.1	53.4 \pm 0.7 ●	17.6 \pm 1.6

Table 9. Results for the EFCIR-U scenario (b). **1st** ●, **2nd** ● and **3rd** ● best metrics are marked accordingly.

Method	A_0	A_{10}	A_{20}	A_{40}	A_{60}	A_{80}	A_{99}	Avg. $A \uparrow$	Avg. $f \downarrow$
WA	67.2 \pm 1.8	52.4 \pm 1.2	50.6 \pm 1.1	48.6 \pm 1.1	47.9 \pm 1.6	47.5 \pm 0.8	46.8 \pm 1.6	49.2 \pm 0.7	18.0 \pm 1.6
FT	63.2 \pm 4.3	25.3 \pm 4.5	29.2 \pm 3.0	32.6 \pm 3.2	35.4 \pm 1.5	37.2 \pm 0.9	39.3 \pm 2.7	34.2 \pm 2.0	29.0 \pm 2.6
FZ	64.2 \pm 4.3	41.3 \pm 3.4	39.9 \pm 3.0	41.0 \pm 3.5	41.0 \pm 3.2	41.5 \pm 2.9	40.9 \pm 2.8	41.7 \pm 3.1	22.5 \pm 1.9
EWC	63.2 \pm 4.3	38.6 \pm 3.5	39.6 \pm 4.6	44.2 \pm 4.1	46.3 \pm 3.2	48.1 \pm 3.0	49.9 \pm 3.2	45.5 \pm 3.2	17.8 \pm 1.8 ●
MAS	63.2 \pm 4.3	39.7 \pm 3.5	41.6 \pm 3.1	45.9 \pm 3.7	49.0 \pm 2.1 ●	49.9 \pm 1.2 ●	51.9 \pm 2.0 ●	47.1 \pm 2.3 ●	16.1 \pm 2.1 ●
LwF	63.2 \pm 4.3	44.8 \pm 2.0	42.9 \pm 2.1	42.7 \pm 1.6	42.6 \pm 0.7	42.0 \pm 1.9	41.0 \pm 2.2	43.5 \pm 0.8	19.8 \pm 4.1
PASS	68.9 \pm 2.0	36.7 \pm 1.9	33.6 \pm 1.9	31.0 \pm 1.2	28.9 \pm 1.9	24.4 \pm 2.4	18.9 \pm 2.4	30.6 \pm 1.4	38.3 \pm 1.6
PRAKA	68.2 \pm 2.2	50.7 \pm 2.7 ●	48.6 \pm 1.7 ●	47.3 \pm 2.5 ●	41.6 \pm 4.3	32.3 \pm 3.7	28.0 \pm 3.6	42.6 \pm 3.2	25.6 \pm 1.9
IL2A	64.4 \pm 4.1	31.9 \pm 2.4	29.7 \pm 2.6	26.9 \pm 3.8	25.2 \pm 2.6	22.3 \pm 2.6	20.2 \pm 2.7	27.2 \pm 2.5	37.2 \pm 1.7
SSRE	52.9 \pm 4.1	27.2 \pm 2.4	25.5 \pm 1.7	25.2 \pm 1.6	24.9 \pm 2.3	24.9 \pm 3.2	25.4 \pm 1.5	26.5 \pm 2.2	26.4 \pm 2.1
FeTrIL	70.7 \pm 1.5 ●	49.1 \pm 0.9	47.4 \pm 0.5	45.1 \pm 1.6	45.5 \pm 1.8	45.2 \pm 1.6	45.0 \pm 1.5	46.9 \pm 0.9	23.8 \pm 1.2
$Horde_m$	72.0 \pm 0.8 ●	52.2 \pm 1.1 ●	53.0 \pm 0.6 ●	54.0 \pm 0.9 ●	54.0 \pm 1.2 ●	55.1 \pm 0.8 ●	54.6 \pm 0.7 ●	54.3 \pm 0.4 ●	17.7 \pm 1.0 ●
$Horde_c$	71.7 \pm 0.8 ●	52.5 \pm 0.7 ●	52.1 \pm 0.7 ●	51.7 \pm 0.7 ●	52.7 \pm 1.4 ●	53.2 \pm 0.8 ●	53.7 \pm 0.4 ●	53.1 \pm 0.4 ●	18.5 \pm 1.1

Table 10. Results for the EFCIR-B scenario (c). **1st** ●, **2nd** ● and **3rd** ● best metrics are marked accordingly.

Leveraging Intermediate Representations for Better Out-of-Distribution Detection

Gianluca Guglielmo¹

Marc Masana^{1,2}

¹Institute of Visual Computing, TU Graz

²SAL Dependable Embedded Systems, Silicon Austria Labs

{guglielmo, mmasana}@tugraz.at

Abstract

In real-world applications, machine learning models must reliably detect Out-of-Distribution (OoD) samples to prevent unsafe decisions. Current OoD detection methods often rely on analyzing the logits or the embeddings of the penultimate layer of a neural network. However, little work has been conducted on the exploitation of the rich information encoded in intermediate layers. To address this, we analyze the discriminative power of intermediate layers and show that they can positively be used for OoD detection. Therefore, we propose to regularize intermediate layers with an energy-based contrastive loss, and by grouping multiple layers in a single aggregated response. We demonstrate that intermediate layer activations improves OoD detection performance by running a comprehensive evaluation across multiple datasets.

1. Introduction

When a model is exposed to data which does not belong to the distribution it was originally trained on, it is desirable that it can detect it and respond appropriately. Therefore, it is beneficial for a machine learning framework to include an *Out-of-Distribution* (OoD) detection mechanism, especially in real-world scenarios. Without it, the model might produce unreliable or even dangerous outputs when confronted with data from an unfamiliar distribution, leading to potential failures in critical applications such as autonomous driving, healthcare, or financial systems [1, 42]. Deep neural networks perform well in many applications but can be overly confident with unseen classes [30]. A key feature would be the ability to avoid providing (over-confident) predictions for unknown classes. Implementing this safety mechanism should not interfere with the intended tasks of the model, such as correctly classifying the samples from the *In-Distribution* (ID) data [42]. However, achieving a balance between ID performance and OoD detection, presents significant challenges. Furthermore, OoD detection mechanisms ought to perform efficiently, without imposing an excessive computational

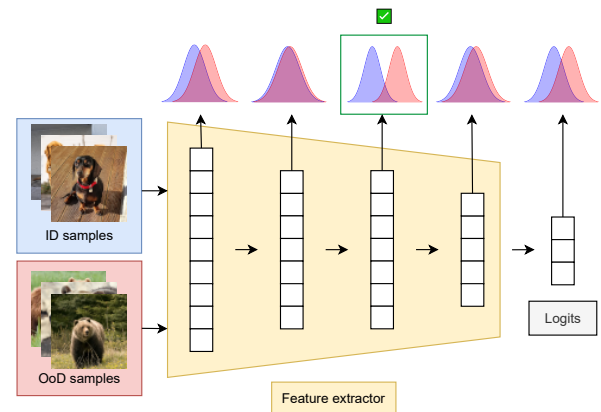


Figure 1. Intermediate representations are often more informative than the logits when dealing with OoD detection.

overhead or diminishing the capacity of the model when performing on the original task. Although recent advances have lead to promising strategies [25, 27, 42], developing methods that achieve this dual goal remains a pressing challenge for the design of trustworthy, scalable AI systems [7].

Building on these challenges, deep learning models have emerged as a powerful solution, becoming the preferred framework for constructing complex training pipelines [9]. These models address the need for effective OoD detection by leveraging their hierarchical architectures, which enable the learning and encoding of *mid-level features* [31], which are representations that bridge low-level patterns such as edges and textures to high-level abstract feature maps such as object parts and semantic categories [12]. In computer vision, these features are inherently diverse, capturing the hierarchical nature of the input data. This diversity not only makes them able to generalize to tasks within the original in-distribution but also highly transferable to new, related tasks. This has been shown within transfer learning scenarios [11].

For OoD detection, most methods rely on penultimate layer embeddings or on the logits of the model [42]. The potential of leveraging ensembles of intermediate layer embeddings remains under-explored [24]. We argue that

mid-level features alone can act as reliable stand-alone OoD-indicators. For instance, inputs with semantically unrelated characteristics compared to the training data may trigger unusual activations in specific layers, serving as an early warning of abnormality.

We pose that specific hidden layers can be effectively isolated and used to enhance OoD detection, performing better than the final layer. However, leveraging these intermediate representations may yield different results depending on the type of shift from ID data — whether semantic or covariate. Also, this may result in varying effects depending on how close or distant the ID and OoD distributions are.

Building on this insight, we test with an aggregated approach, which leverages hidden-layer information in a layer-agnostic manner. This method avoids relying on specific layers, which enables a more robust and generalized use of the network’s intermediate representations for OoD tasks. Further, we also propose an approach that regularizes selected hidden layers through an energy-based contrastive loss, improving OoD detection by leveraging their intermediate representations. The goal is to promote the information encoded in the hidden spaces to be distributed such that OoD detection is more efficient, and without disrupting the ID task performance.

Therefore, our contributions are summarized as:

- we establish that the embeddings of hidden layers are valuable for OoD detection,
- we introduce a layer-agnostic aggregated (Ag-EBO) approach that leverages intermediate representations,
- we propose a modular strategy to enhance robustness by regularizing specific layers (R-EBO).

The article is structured as follows: Sec. 2 presents a current overview of the OoD field, while Sec. 3 introduces the preliminaries needed for the study in Sec. 4, which shows that intermediate layers contain useful information for the OoD detection task. An overview of the proposed ways of exploiting this capabilities, with detailed results is presented in Sec. 5. Finally, in Sec. 6 we discuss some limitations of the proposed approaches and the main take-aways.

2. Related Work

In this paper, we concentrate on two main families of Out-of-Distribution methods: *post-hoc* and *training-based* [25, 42].

Post-hoc methods are applied after the model has been trained and typically involve analyzing its predictions or intermediate representations to identify whether an input is OoD [14, 17, 41]. These methods often focus on computational efficiency and adaptability to pre-trained models, as they avoid retraining [25].

Training-based methods modify the training process, sometimes completely restructuring the model to accommodate OoD detection [6, 20, 34]. These methods often

come at the cost of higher training complexity, and might dilute the efforts to obtain an optimal ID training accuracy [25, 28]. Additionally, exposure to outliers (real or generated) can be done to improve generalization [39].

Baselines. A classic baseline for OoD is considered to be Maximum Softmax Probability (MSP) [17], a simple approach that relies on the logit scores to identify OoD samples. However, a major limitation of this approach is the tendency of models to produce overconfident predictions on anomalous data, leading to poor performance [14]. Temperature scaling [14] is a simple post-hoc way of tackling the overconfidence issue, where logits are scaled by a temperature T , but its results are not optimal [43].

OoD and intermediate layers. Some methods leverage intermediate embeddings within the network. However, most do it to refine the head’s detection capabilities, rather than for direct OoD detection. ASH [8] enhances the network’s OoD detection capabilities through activation masking of hidden layers. Similarly, ReAct [32] proposes to rectify the embeddings of the penultimate layer to reduce overconfidence. However, despite leveraging intermediate embeddings to an extent, the final detection decisions in both methods rely solely on the output logits. Mahalanobis distance-based method (MDSEns) [24] uses features from hidden layers to compute distances from the known distribution. However, this approach relies on the assumption that the class-conditional distributions of hidden layer features are Gaussian, which may not hold true for complex datasets and deep network architectures [36]. Head2Toe [11] leverages intermediate representations by training a classifier head on concatenated embeddings from multiple hidden layers to improve generalization during *transfer learning*. This enables the refinement of existing OoD detection techniques through the utilization of hidden layer structures.

3. Out-of-Distribution Detection

3.1. Problem statement

In Out-of-Distribution (OoD) detection, the objective is to differentiate between samples generated by the same distribution as the in-distribution dataset, \mathcal{D}_{in} , and those originating from a different, out-of-distribution dataset, \mathcal{D}_{out} . Due to the complexity and variance of image-based data, the concept of the amount of *out-of-distributionness* of samples is inherently challenging to define. However, two primary types of distributional shifts are commonly identified [35]:

- **Semantic (or Concept) shifts:** they arise when new classes appear at test time. For instance, encountering an image of a dog after the model has been trained on pictures of cats and mice.
- **Covariate shifts:** occur when the style or attributes of samples change within the same class. Examples include image corruptions [16], such as artifacts, blurs or noise, and domain changes [18, 38], such as shifting from natural photographs to artistic paintings.

Both semantic and covariate shifts can occur with varying levels of severity depending on the problem, and can also appear entangled within a distribution shift. Given a fixed \mathcal{D}_{in} , we refer to *near* and *far* OoD datasets as those that are semantically closer to or further from it, respectively.

Moreover, depending on the OoD detection application, different shifts might be considered within the spectrum that comprises between novelty and anomaly detection [28]. The first relates to distribution shift that might need to be explicitly added to the model, while the second is usually added in a more implicit way, in order to efficiently use the capacity of the model. In this paper, we do not distinguish samples based on the suitability for further learning, but instead aim to analyze these shifts from a perspective of distribution similarity.

Terminology. Consider a neural network $f(\mathbf{x}; \theta)$ with input \mathbf{x} and parameters θ , and trained to classify C classes. The architecture of the network is defined as a series of L layers with intermediate functions such that:

$$y = f(\mathbf{x}; \theta) = (f_L^{\theta_L} \circ f_{L-1}^{\theta_{L-1}} \circ \dots \circ f_1^{\theta_1})(\mathbf{x}),$$

where the output y is a vector of C logits representing the unnormalized prediction over the classes. Therefore, the intermediate representations or embeddings of a given layer l are defined as:

$$\mathbf{a}_l = (f_l \circ \dots \circ f_1)(\mathbf{x}).$$

To determine whether an input \mathbf{x} belongs to \mathcal{D}_{in} or \mathcal{D}_{out} , a score function $\mathcal{S}(\mathbf{x})$, is usually derived from the neural network. This score reflects the confidence of the model in the input belonging to the expected in-distribution. A threshold T is applied to classify the input such that:

$$g(\mathbf{x}) = \begin{cases} \mathbf{x} \in \mathcal{D}_{in} & \text{if } \mathcal{S}(\mathbf{x}) \geq T \\ \mathbf{x} \in \mathcal{D}_{out} & \text{if } \mathcal{S}(\mathbf{x}) < T. \end{cases}$$

The threshold can be adjusted depending on the desired balance between sensitivity and specificity for OoD detection.

Metrics. In order to evaluate the strength of a method, two essentials metrics are AUROC, the Area Under the Receiver Operating Characteristic (the higher the better) and FPR@TPR95, the False Positive Rate when the True Positive Rate is 95% (the lower the better).

3.2. Energy-based out-of-distribution detection.

Energy-based models [23] have demonstrated to be effective as post-hoc OoD detectors. The *free energy function* $E(\mathbf{x}; \mathbf{f})$ is defined as:

$$E(\mathbf{x}; \mathbf{f}) = -T \log \sum_{c=1}^C e^{f^c(\mathbf{x})/T}, \quad (1)$$

where T is the temperature, for temperature scaling [14]. When $T = 1$, it simplifies to the negative log of the denominator of the softmax function, which represents the

normalization factor in the softmax computation. In this case, the energy function effectively captures the aggregate contribution of all logits, weighted by their exponential, to produce a measure of confidence over the entire output distribution. The Energy-Based OoD (EBO) [26] detection approach uses the free energy associated to each input to determine whether it is ID or OoD, where the higher the energy is, the more likely the sample is OoD. JEM [13] is another energy-based approach that improves the calibration (the mismatch between accuracy and confidence) of the model.

4. OoD with Intermediate Layers

4.1. Motivation

As data moves through the trained layers of the network, the represented features become more complex, from edges and simple texture patterns to higher-level representations or combinations of intermediate features [31]. Our assumption is that the use of these intermediate representations can improve out-of-distribution detection. Therefore, we take EBO [26] as a starting point and analyze how discriminative the different layers of the model are for OoD detection. To quantify the capacity of hidden layers in the OoD task, we introduce a hypothetical method called *Best Hidden Layer* (BHL), which utilizes an oracle to identify the optimal hidden layer for OoD detection. Therefore, since it requires access to the distribution ground truth, it is proposed as an a-posteriori analysis strategy.

Following classic setups, we train a classification model on \mathcal{D}_{in} , using the standard *cross-entropy* loss \mathcal{L}_{CE} . Then, we evaluate on test data from both \mathcal{D}_{in} and \mathcal{D}_{out} , extracting the embeddings from the intermediate layers for each sample. Here, the free energy from Eq. (1) is a natural candidate to use on the logits. However, the function can also take the embeddings \mathbf{a}_l from any other layer l . Thus, we propose to extract the energy score:

$$E_l(\mathbf{x}) = -T \log \sum_i e^{a_i^l(\mathbf{x})/T}, \quad (2)$$

where the unit indices i correspond to the output of the l -th layer.

We extract and analyze the energy of each layer, regardless of its type, such as convolutional, batch normalization, or fully connected. We observe that certain intermediate layers consistently outperform the network logits from the original EBO approach. This effect is shown in Figure 2 for semantic shift, which presents the AUROC scores evaluated across all layers of a ResNet18 [15] for CIFAR-10 [22] as \mathcal{D}_{in} and near and far OoD as \mathcal{D}_{out} . Some high-performing layers exhibit unexpected behavior by assigning lower energy values to \mathcal{D}_{out} samples instead of \mathcal{D}_{in} samples. This leads to two possibilities: assigning OoD to lower energy samples or to higher energy samples. Among the two, the ‘‘correct’’ possibility is reflected in the reported results.

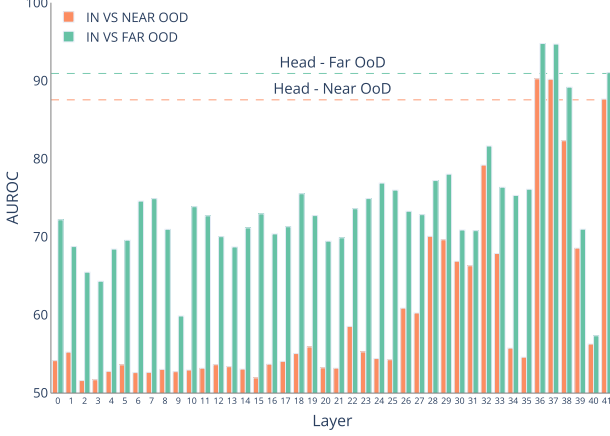


Figure 2. AUROC scores for OoD detection for each intermediate layer of ResNet18 are presented. The network is pretrained on CIFAR-10 (\mathcal{D}_{in}) and evaluated against the corresponding \mathcal{D}_{out}^{near} and \mathcal{D}_{out}^{far} datasets. Results are averaged across datasets in both categories.

Covariate shift OoD detection also shows significant improvement when considering intermediate layers rather than relying solely on the network’s output logits. To test it, we look at the performance of different layers when the OoD represents the in-distribution shifted by different corruptions (CIFAR-10-C [16], see Sec. 5.1). Figure 3 shows that throughout the depth of the network, several layers outperform yet again the head. Initial layers, which provide low-level features such as edges or local histogram projections, seem to be good candidates for OoD detection when covariate shift is present, since it represents a transformation on the in-distribution.

Despite the clear benefits from using some of the layers, determining which one to use for OoD detection under different shifts is still challenging due to different \mathcal{D}_{out} distributions or modes having a tendency to elicit the strongest responses in different layers. This variability means that no single layer is universally optimal for detecting all types of OoD inputs effectively. It must be noted that, on average for semantic shifts, the optimal layers are observed to reside more towards the later layers of the network (see Fig. 2). However, this is not enough to identify a good one-fits-all layer, or to find a straightforward selection criteria. We try to circumvent this issue by proposing two strategies to leverage the information from the intermediate layer representation spaces:

- aggregating all intermediate responses into a single unified response (described in Sec. 4.2);
- strictly regularize selected layers to enforce generalization over different distributions (described in Sec. 4.3).

4.2. Energy aggregation (Ag-EBO)

To develop a fully layer-agnostic post-hoc method that leverages all the potential from intermediate embeddings, we propose to aggregate the energy values extracted from all L layers simultaneously. Thus, for each input \mathbf{x} , we

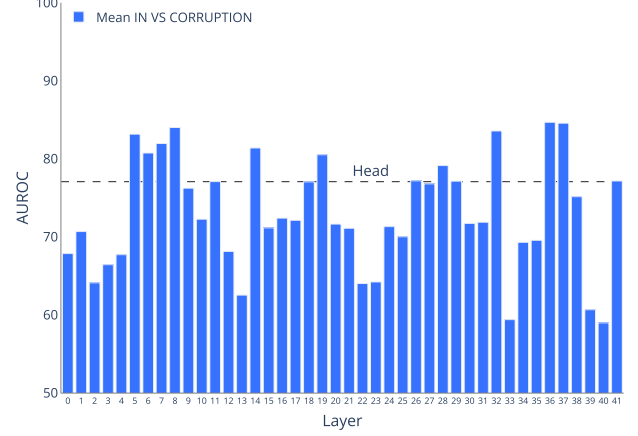


Figure 3. AUROC scores for each intermediate layer of ResNet18 pretrained on CIFAR-10 as \mathcal{D}_{in} and evaluated against different corruptions (CIFAR-10-C). Results are averaged over all corruption types and seeds.

construct a vector of energies:

$$\mathbf{E}(\mathbf{x}) = (E_1(\mathbf{x}), \dots, E_L(\mathbf{x})),$$

which groups the energy contributions of each layer into a unified representation. The dimension of this vector is significantly smaller than the total hidden dimension of the network, making it scalable and suitable for use with most common OoD methods. However, for the intermediate layer to be considered, it is desirable that it offers better results than just relying on the logits or on the embeddings from the penultimate layer.

We tested with some straightforward approaches from literature, presented in the next paragraphs. Two of the following three methods need a reference for the ID data, therefore we use the set of energies $\tilde{\mathbf{E}} = \mathbf{E}_{in}^{\text{train}} = \{\mathbf{E}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{D}_{in}^{\text{train}}\}$, extracted from $\mathcal{D}_{in}^{\text{train}}$.

Mahalanobis distance. The score $\mathcal{S}_{MD}(\mathbf{x})$ depends on the Mahalanobis distance [24] of $\mathbf{E}(\mathbf{x})$:

$$\mathcal{S}_{MD}(\mathbf{x}) = \min_{\mu_c \in \tilde{\mathbf{E}}} \sqrt{(\mathbf{E}(\mathbf{x}) - \mu_c)^\top \Sigma_c^{-1} (\mathbf{E}(\mathbf{x}) - \mu_c)},$$

where μ_c and Σ_c are the mean vector and covariance matrix of the energy vectors for class c in $\mathcal{D}_{in}^{\text{train}}$, respectively.

K-nearest neighbor. The score $\mathcal{S}_{KNN}(\mathbf{x})$ is based on the distance of $\mathbf{E}(\mathbf{x})$ to its K nearest neighbors [33]:

$$\mathcal{S}_{KNN}(\mathbf{x}) = \frac{1}{K} \sum_{i=1}^K \|\mathbf{E}(\mathbf{x}) - \mathbf{E}_i\|_2,$$

where $\{\mathbf{E}_1, \dots, \mathbf{E}_K\} \subset \tilde{\mathbf{E}}$ are the K nearest neighbors of $\mathbf{E}(\mathbf{x})$ in the in-distribution training set, measured using the Euclidean distance.

Reconstruction Error. The score $\mathcal{S}_{VAE}(\mathbf{x})$ is computed as the reconstruction error of $\mathbf{E}(\mathbf{x})$ using a small Variational Autoencoder [21]:

$$\mathcal{S}_{VAE}(\mathbf{x}) = \|\mathbf{E}(\mathbf{x}) - \hat{\mathbf{E}}(\mathbf{x})\|_2,$$

	CIFAR-10 [22]	CIFAR-100 [22]	ImageNet200 [4]	ImageNet [4]
Architecture	RESNET18 [30]	RESNET18	RESNET18	RESNET50 [30]
Input Size	$32 \times 32 \times 3$	$32 \times 32 \times 3$	$224 \times 224 \times 3$	$224 \times 224 \times 3$
Near-OoD	CIFAR-100 TINYIMAGENET [4]	CIFAR-10 TINYIMAGENET	SSB-HARD [43] NINCO [2]	SSB-HARD NINCO
Far-OoD	TEXTURE [3] MNIST [5] SVHN [29] PLACES365 [44]	TEXTURE MNIST SVHN PLACES365	TEXTURE INATURALIST [19] OPENIMAGEO [37] -	TEXTURE INATURALIST OPENIMAGEO -
Corruptions	CIFAR-10-C [16]	-	-	-

Table 1. Setup description for each ID dataset.

	CIFAR-100	TIN	Near OoD	MNIST	Places365	SVHN	Texture	Far OoD
EBO [26]	86.36	88.80	87.58	94.32	89.25	91.79	89.47	91.21
BHL	88.23	92.26	90.25	99.89	92.13	98.46	93.5	96.00
MDSEns [24]	61.29	59.57	60.43	99.17	66.56	77.40	52.47	73.90
Ag-EBO w/ MD	66.03	67.03	66.53	99.05	63.73	94.25	93.32	87.59
Ag-EBO w/ KNN	83.69	86.5	85.09	92.81	86.01	89.64	87.46	88.98
Ag-EBO w/ VAE	80.42	83.11	81.77	89.31	83.27	88.25	84.13	86.24

Table 2. AUROC scores of MDSEns, EBO, BHL and three aggregation methods with CIFAR-10 as \mathcal{D}_{in} , averaged over 3 runs.

where $\hat{\mathbf{E}}(\mathbf{x})$ is the reconstruction of $\mathbf{E}(\mathbf{x})$. Higher reconstruction error indicates that the input is likely to be out-of-distribution.

4.3. Energy regularization (R-EBO)

Regularizing intermediate layers directly provides an effective approach to addressing the intermediate layer selection problem. Ideally, by enforcing a strong energy-based discriminative behavior within the hidden layers, we promote their reliability, allowing them to be used confidently without additional selection mechanisms.

EBO [26] introduces an energy-bounded learning loss $\mathcal{L}_{\text{energy}}$ to push the network to assign low energy values to ID samples (and viceversa for OoD). Since their approach operates at the logits level, this loss is applied exclusively to the model’s head. In contrast, our proposed strategy extends the scope of this loss by applying it to each hidden convolutional layer during training, computing and back-propagating all the losses simultaneously. Given an ID dataset $\mathcal{D}_{in}^{\text{train}}$ and an OoD seen dataset $\mathcal{D}_{out}^{\text{train}}$ (for outlier exposure), the energy regularization loss for the l -th hidden layer is defined as:

$$\mathcal{L}_{\text{energy},l} = \mathbb{E}_{\mathbf{x}_{in} \sim \mathcal{D}_{in}^{\text{train}}} [\max(0, E_l(\mathbf{x}_{in}) - m_{in})]^2 + \mathbb{E}_{\mathbf{x}_{out} \sim \mathcal{D}_{out}^{\text{train}}} [\max(0, m_{out} - E_l(\mathbf{x}_{out}))^2], \quad (3)$$

where m_{in} and m_{out} are two margins, serving as the upper bound for the energy of the ID data and the lower bound for the energy of the seen OoD data, respectively. We

define the total loss as:

$$\mathcal{L}_{\text{R-EBO}} = \sum_{l=1}^L \mathcal{L}_{\text{energy},l}, \quad (4)$$

where the same constant margin values for m_{in} and m_{out} are used across all layers, although each can be explored independently. In the original EBO paper [26], $\mathcal{L}_{\text{EBO}} = \mathcal{L}_{\text{energy},E_L}$, where L is the last layer of the network. Furthermore, the decision to reduce the free ID energy and increase the OoD energy in intermediate layers is a design choice. Alternative regularization strategies can also be considered.

5. Experimental results

5.1. Implementation details

Datasets. The datasets used in this study were selected based on the guidelines of the OpenOoD benchmark [43], which offers a comprehensive and well-documented collection of state-of-the-art (SoTA) methods across various OoD scenarios. Also, the results presented here have been extracted from its continuously updated report, to ensure alignment with the latest developments in the field. For each \mathcal{D}_{in} , the OpenOoD benchmark defines a set of semantically *near* and *far* OoD datasets from it Table 1. Additionally, we tested the response to covariate shift from CIFAR-10 with the corruptions dataset CIFAR-10-C [16]. This is a dataset consisting of corrupted versions of CIFAR-10 images, which serves as a

common benchmark for evaluating robustness to covariate shifts. It includes a variety of corruption types, such as noise, blur, and weather distortions, applied at varying levels of severity.

Architectures. To keep the consistency with OpenOoD evaluations, the main results have been calculated using the same architectures used in the benchmark, shown in Tab. 1. We also evaluate on a non-residual based convolutional neural network, EFFICIENTNET-B7, for which we select convolutional, fully-connected, batch normalization and average pooling layers. Finally, following recent trends in machine learning, we evaluate ViT-B-16 [10], a transformer-based [40] architecture. ViTs utilize multi-head self-attention layers, and their feed-forward sub-layers consist of fully-connected layers. Our experiments focus on the selection of these fully-connected layers for BHL.

Training. OpenOoD provides three pretrained ResNet18 checkpoints for CIFAR-10, CIFAR-100, and IMAGENET200 as \mathcal{D}_{in} , and a single pretrained ResNet50 checkpoint for IMAGENET, all trained using standard SoftMax loss. Additionally, we trained 3 checkpoints for both CIFAR-10 and CIFAR-100 as \mathcal{D}_{in} using the hidden regularization approach.

5.2. Analysis of OoD with intermediate layers

In Table 2, CIFAR-10 is selected as \mathcal{D}_{in} . EBO refers to the standard energy-based OoD detection mechanism applied directly at the logit level, while BHL shows the energy-based OoD detection using the best performing hidden layer. The results presented for BHL are averaged across the best hidden layer identified in each run, which tends to slightly vary between runs. For every \mathcal{D}_{out} the results are strongly improved by (at least) one hidden layer’s response. It is important to mention that the results presented only consider the internal behavior of the network, while an algorithm which correctly weighs the importance of a layer for OoD detection would also take the head of the model into consideration, potentially merging the best results of the two rows.

Energy aggregation. The last rows of Table 2 present the results of the aggregation methods (Ag-EBO) proposed in Section 4. The row above displays the results of MDSEns [24], taken from the OpenOoD benchmark [43]. Each of our proposed aggregation methods achieves higher AUROC compared to MDSEns [24], an ensemble method that exploits Mahalanobis distance on hidden layers. The lower results for MDSEns might be related to their assumption of class-conditional distribution of the hidden features being Gaussian. \mathcal{D}_{out}^{far} datasets, such as MNIST, SVHN, and TEXTURE, demonstrate improved performance with the KNN aggregation approach compared to EBO. However, none of these methods are robust enough on average to consistently outperform relying exclusively on the head logits. This indicates that the layer-selection problem remains unsolved and cannot yet be effectively simplified into an aggregation mechanism.

	CIFAR-10		CIFAR-100	
	Far	ID Acc.	Far	ID Acc.
EBO [26]*	84.86	82.33	67.86	54.83
BHL*	90.42	82.33	86.98	54.83
R-EBO*	98.48	78.2	94.06	50.05

Table 3. AUROC scores of EBO, BHL and R-EBO with CIFAR-10 as \mathcal{D}_{in} , averaged over multiple runs. EBO and BHL exploit identical checkpoints, retrained (*) for direct comparability with R-EBO.

Dataset	EBO [26]	BHL	R-EBO
BRIGHTNESS	56.51	82.98	79.8
CONTRAST	92.39	99.92	96.99
DEFOCUS BLUR	84.65	97.26	85.44
ELASTIC	73.24	87.36	85.11
FOG	71.3	96.7	94.38
FROST	76.83	91.4	91.08
GAUSSIAN BLUR	89.8	98.86	75.96
GAUSSIAN NOISE	84.39	99.65	99.16
GLASS BLUR	85.77	88.45	98.13
IMPULSE NOISE	89.04	99.98	97.71
JPEG	73.33	87.87	61.47
MOTION BLUR	75.78	93.51	72.77
PIXELATE	80.02	94.17	99.66
SATURATE	57.47	90.97	81.8
SHOT NOISE	84.93	99.38	98.78
SNOW	71.85	89.11	74.95
SPATTER	71.0	90.33	83.69
SPECKLE NOISE	85.29	98.96	98.66
ZOOM BLUR	79.36	96.61	66.11

Table 4. AUROC scores of EBO, BHL, and R-EBO with CIFAR-10 as \mathcal{D}_{in} against corruption datasets.

Energy regularization. Table 3 presents the results of regularization against other SoTA methods that exploit \mathcal{D}_{out}^{seen} . The margin values are set to $m_{in}=-25$ and $m_{out}=-7$, following the original EBO setup [26]. In order to test the trade-off in hidden layer regularization compared to a completely post-hoc hidden layer analysis, we selected CIFAR-10 and CIFAR-100 as \mathcal{D}_{in} and IMAGENET as \mathcal{D}_{out}^{seen} . We then trained 5 runs using only \mathcal{L}_{CE} , and 5 runs using $\mathcal{L}_{CE} + \mathcal{L}_{R-EBO}$. We opted not to use the checkpoints given by OpenOoD to guarantee a fair comparison between the two losses. Therefore, the EBO results are not comparable with the ones presented in other tables, and are marked with (*) accordingly. Moreover, only results related to Far-OoD are presented, since Near-OoD includes TIN, which is based on IMAGENET.

As expected, the regularization of intermediate layers

		CIFAR-10		CIFAR-100		ImageNet-200		ImageNet-1K	
		Near	Far	Near	Far	Near	Far	Near	Far
ResNet18/50	EBO	87.58	91.21	80.91	79.77	82.50	90.86	75.89	89.47
	BHL	90.25	96.00	71.57	86.08	86.72	76.13	79.04	89.75
EfficientNet-B7	EBO	97.39	98.91	87.46	86.91	75.02	86.53	65.16	81.65
	BHL	87.43	99.74	84.21	99.80	78.83	93.02	85.24	94.49
ViT-B-16	EBO	90.91	93.9	88.81	87.23	69.72	83.49	62.93	78.71
	BHL	79.38	96.14	81.38	97.98	62.19	81.40	74.06	88.43

Table 5. EBO and BHL compared on different models.

strongly improves the OoD detection capabilities of the model on both cases. However, this comes at the cost of a slight decrease in ID accuracy, due to the additional \mathcal{L}_{R-EBO} loss.

Covariate shift. Table 4 presents the detailed OoD results of CIFAR-10 against every corruption type present in CIFAR-10-C. As with the semantic shift, we observe that covariate shift is better identified by the hidden layers rather than by the final logits. Table 4 also presents R-EBO results under covariate shift conditions, evaluated using the same checkpoints from Table 3. The findings suggest that regularizing layers with a semantically distinct \mathcal{D}_{out}^{seen} does not consistently enhance the identification of covariate shift.

5.3. Analysis on different architectures

Table 5 presents the complete results for EBO and BHL, averaged over multiple runs, using RESNET18/50, EFFICIENTNET-B7, and ViT-B-16 as backbones. The findings are consistent with earlier observations: BHL improves performance in most setups, except for certain Near OoD cases.

6. Discussion and Limitations

Our findings show that intermediate representations are capable of discriminating out-of-distribution samples better than the logits. Both semantic, in the form of unseen classes, and covariate shift, in the form of image corruptions, are strongly captured by intermediate layers. However, a robust selection criterion for which layer to use is still an open question, since the proposed aggregation method underperforms compared to simpler logit-based alternatives.

Regularization of the intermediate layer’s energies improves the results even further, albeit with a trade-off in ID accuracy. We suspect that the influence of \mathcal{D}_{out}^{seen} leads to sub-optimal filters for the discrimination of ID classes, thus motivating further research involving regularization which exploits \mathcal{D}_{in} only. Additionally, regularization using synthetic generated data [45] applied to intermediate layers could also be a promising direction, as it would reduce dependence on specific datasets, promote privacy-preservation, and enhance the generalization.

Finally, the findings on this paper pave the way for real-time optimized out-of-distribution detection, enabling the identification of OoD samples in earlier layers during network propagation. By detecting such samples promptly, the system can flag them and halt further processing, reducing computational overhead and improving efficiency.

Acknowledgements

Gianluca Guglielmo acknowledges the support of KAI GmbH and Infineon Technologies Austria. Marc Masana acknowledges the support by the “University SAL Labs” initiative of Silicon Austria Labs (SAL).

References

- [1] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv e-prints*, pages arXiv–1606, 2016. 1
- [2] Julian Bitterwolf, Maximilian Mueller, and Matthias Hein. In or out? fixing imagenet out-of-distribution detection evaluation. In *ICML*, 2023. 5
- [3] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5
- [5] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 5
- [6] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv e-prints*, pages arXiv–1802, 2018. 2
- [7] Natalia Díaz-Rodríguez, Javier Del Ser, Mark Coeckelbergh, Marcos López de Prado, Enrique Herrera-Viedma, and Francisco Herrera. Connecting the dots in trustworthy artificial intelligence: From ai principles, ethics, and key requirements to responsible ai systems and regulation. *Information Fusion*, 99:101896, 2023. 1
- [8] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely Simple Activation Shaping for Out-of-Distribution Detection. *arXiv e-prints*, art. arXiv:2209.09858, 2022. 2

- [9] Shi Dong, Ping Wang, and Khushnood Abbas. A survey on deep learning and its applications. *Computer Science Review*, 40:100379, 2021. 1
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 6
- [11] Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C Mozer. Head2toe: Utilizing intermediate representations for better transfer learning. In *International Conference on Machine Learning*, pages 6009–6033. PMLR, 2022. 1, 2
- [12] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016. <http://www.deeplearningbook.org>. 1
- [13] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2019. 3
- [14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 2, 3
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3
- [16] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. 2, 4, 5
- [17] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv e-prints*, pages arXiv–1610, 2016. 2
- [18] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8340–8349, 2021. 2
- [19] Grant Van Horn, Oisín Mac Aodha, Yang Song, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist challenge 2017 dataset. *CoRR*, abs/1707.06642, 2017. 5
- [20] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020. 2
- [21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 4
- [22] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 3, 5
- [23] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fuyang Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 3
- [24] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 1, 2, 4, 5, 6
- [25] Jiashuo Liu, Zheyang Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv e-prints*, pages arXiv–2108, 2021. 1, 2
- [26] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020. 3, 5, 6
- [27] Shuo Lu, YingSheng Wang, LuJun Sheng, AiHua Zheng, LinXiao He, and Jian Liang. Recent advances in ood detection: Problems and approaches. *arXiv e-prints*, pages arXiv–2409, 2024. 1
- [28] Marc Masana, Idoia Ruiz, Joan Serrat, Joost van de Weijer, and Antonio M Lopez. Metric learning for novelty and anomaly detection. In *British Machine Vision Conference (BMVC)*, 2018. 2, 3
- [29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bisaccho, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011, 2011. 5
- [30] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. 1, 5
- [31] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014. 1, 3
- [32] Yiyao Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021. 2
- [33] Yiyao Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR, 2022. 4
- [34] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 33:11839–11852, 2020. 2
- [35] Junjiao Tian, Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Exploring covariate and concept shift for detection and calibration of out-of-distribution data. *arXiv e-prints*, pages arXiv–2110, 2021. 2
- [36] Aishwarya Venkataramanan, Assia Benbihi, Martin Laviale, and Cédric Pradalier. Gaussian latent representations for uncertainty estimation using mahalanobis distance in deep classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4488–4497, 2023. 2
- [37] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4911–4920, 2022. 5
- [38] Hongjun Wang, Sagar Vaze, and Kai Han. Dissecting out-of-distribution detection and open-set recognition: A crit-

- ical analysis of methods and benchmarks. *International Journal of Computer Vision*, pages 1–26, 2024. [2](#)
- [39] Qizhou Wang, Zhen Fang, Yonggang Zhang, Feng Liu, Yixuan Li, and Bo Han. Learning to augment distributions for out-of-distribution detection. *Advances in neural information processing systems*, 36:73274–73286, 2023. [2](#)
- [40] A Waswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A Gomez, L Kaiser, and I Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017. [6](#)
- [41] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International conference on machine learning*, pages 23631–23644. PMLR, 2022. [2](#)
- [42] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024. [1](#), [2](#)
- [43] Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyu Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, et al. Openood v1.5: Enhanced benchmark for out-of-distribution detection. *arXiv e-prints*, pages arXiv–2306, 2023. [2](#), [5](#), [6](#)
- [44] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1452–1464, 2018. [5](#)
- [45] Jianing Zhu, Yu Geng, Jiangchao Yao, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Bo Han. Diversified outlier exposure for out-of-distribution detection via informative extrapolation. *Advances in Neural Information Processing Systems*, 36:22702–22734, 2023. [7](#)

Real-time object detection in diverse weather conditions through adaptive model selection on embedded devices

Mohammad Milad Tufan^{1,2,3} Christian Fruhwirth-Reisinger^{1,2} M. Jehanzeb Mirza^{1,2}
Darko Štern^{2,3}

¹Institute of Visual Computing, Graz University of Technology

²Christian Doppler Laboratory for Embedded Machine Learning, Austria

³AVL List GmbH

m.tufan@student.tugraz.at, {reisinger, mirza}@tugraz.at, darko.stern@avl.com

Abstract

The perception system is a critical component of Advanced Driver Assistance Systems (ADAS) and Automated Driving (AD), playing a pivotal role in reducing traffic accidents caused by human error. For ADAS/AD systems to be seamlessly integrated into everyday life, it is essential to ensure the reliable operation of their perception systems, even under challenging conditions such as adverse weather. This paper presents a novel perception pipeline for real-time object detection with YOLOv3 across diverse weather scenarios. The pipeline incorporates adaptive model selection based on current conditions to optimize detection performance dynamically. To address the computational limitations of embedded systems in constraint environments, we propose a three-step approach: (1) reduction of YOLOv3 complexity using L^1 regularization for feature selection, followed by (2) weight pruning and (3) knowledge distillation to recover precision lost in earlier steps. This results in lightweight models up to 70% smaller than the base model while maintaining high precision through knowledge distillation. Finally, the optimized models are evaluated on resource-constrained embedded devices, including the NVIDIA Jetson AGX Orin, NVIDIA Jetson Nano, and Raspberry Pi 4, demonstrating robust and efficient performance under real-world conditions.

1. Introduction

Advanced Driver-Assistance Systems (ADAS) play a crucial role in enhancing road safety by mitigating risks associated with human error [2], which remains a leading cause of traffic accidents. According to the European Commission’s 2021 accident report [8], approximately 100,000 traffic accidents involving personal injury occurred in the EU, with 20% resulting in fatalities. Human factors such as distraction, fatigue, or delayed reac-

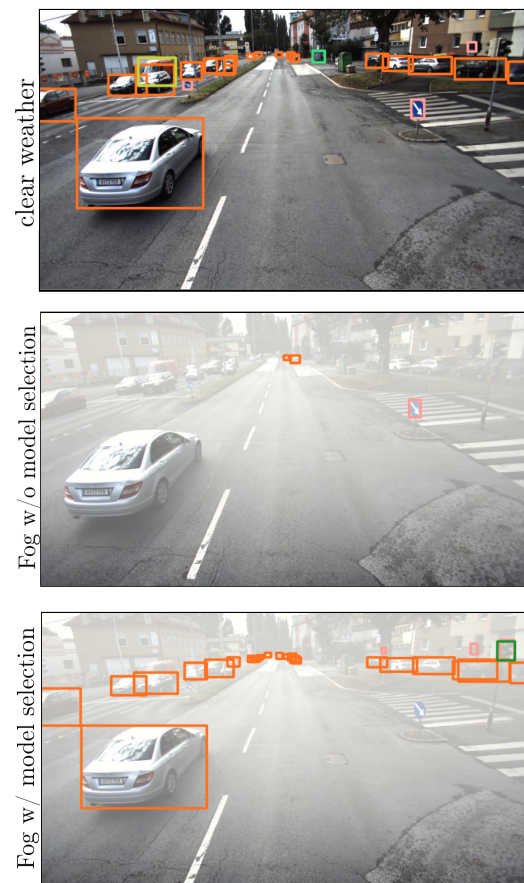


Figure 1. The detection examples demonstrate the need for adaptation in real-time. The second row shows detections with the clear weather model on a foggy image, while the last row shows detections with our adaptive perception pipeline on a foggy image. Fog was injected artificially into the clear weather image.

tions account for a significant proportion of these incidents. ADAS technologies have the potential to prevent many of these accidents or at least reduce their severity,

making their development and deployment critical.

Traditional ADAS functionalities, such as *forward collision warning*, *automatic emergency braking*, and *traffic sign recognition*, rely predominantly on rule-based systems. While effective in specific scenarios, these systems are highly application-specific and lack adaptability to diverse environments or evolving requirements. With the advent of deep learning, ADAS have gained significant versatility and accuracy, enabling tasks such as object detection, scene understanding, and environment perception. These capabilities form the foundation for both ADAS and automated driving (AD), where reliable detection of traffic participants is essential for safe and efficient operation. Despite the advancements brought by deep learning, these methods often require large-scale, meticulously annotated datasets to perform reliably. The process of collecting, storing, and labeling such data poses significant challenges. Memory and processing constraints further complicate the ability to save and review all recorded scenes, particularly in dynamic environments. Determining which scenes should be labeled for training or analysis is a complex task that relies on exhaustive data exploration, increasing time and resource costs.

A practical approach to this challenge is deploying lightweight, real-time object detection systems directly on the recording platforms. These systems serve as an initial filter to pre-select scenes containing relevant objects or events for further processing. By focusing on critical areas of interest, such as scenes with traffic participants or specific environmental conditions, such systems reduce the burden of exhaustive data storage and labeling while ensuring that the most informative samples are identified. Although accuracy is not the primary goal in this context, detectors with higher precision naturally lead to better-informed data selection decisions, ultimately enhancing the performance of subsequent deep learning pipelines.

In this paper, we propose a perception pipeline that dynamically adapts to various weather conditions via model selection and runs in real-time on embedded platforms with constrained resources. In particular, we train a YOLO [26] expert for each weather scenario, *i.e.*, *clear*, *rain*, and *fog* to deal with occurring distribution shifts. While inference, a weather domain classifier decides which model to use. The proposed expert selection design ensures precise detections in dynamic environments, as shown in Fig. 1. To reach real-time performance on embedded devices, we follow two strategies: 1) model pruning and 2) tiny models. In both cases, we perform knowledge distillation from the base model to achieve adequate performance. Our contributions are as follows:

- We propose a real-time perception pipeline, depicted in Fig. 2, deployable on various embedded devices. This pipeline tackles distribution shifts by first recognizing the domain and, secondly, switching to an appropriate expert model.
- With a sparse training and pruning procedure, we reduce model size and complexity to perform real-time per-

ception on edge devices. Afterward, we distill knowledge [15] from the base model to regain the precision lost in the pruning process.

- Finally, we perform exhaustive evaluations in clear and adverse weather conditions and provide a detailed run-time and memory analysis on various edge devices.

2. Related work

Object detection. The localization and classification of objects is a crucial task in many challenging real-world applications like robotics [13, 30] or autonomous driving [1]. It becomes even more challenging when applied in constrained environments like embedded devices [33], especially when real-time processing is required. To that end, object detection has been extensively researched in the past. Examples are EfficientDet [31], DETection TRansformer (DETR) [4], or CenterNet [10]. Object detectors can be separated into two categories: By a two-stage detector using region proposals [12, 27] or by a one-stage detector with a unified network architecture that treats object detection as a regression problem [19, 20]. In our pipeline, we apply the YOLO [26] object detector. It has a lightweight architecture that applies only a single stage and provides satisfactory results. With appropriate optimization, it can run in real time on embedded devices.

Distribution shifts. In Machine Learning, we draw training samples from a distribution p_{train} that we assume to be independent and identically distributed (i.i.d.). Further, we assume our inference data to be drawn from the same or at least very similar distribution $p_{inference}$. However, since not all samples of a highly dynamic environment are known in advance, *e.g.*, training on clear weather and inference on rainy weather, unknown samples inevitably lead to a distribution shift such that $p_{inference} \neq p_{train}$. The need for adaptive model selection arises to handle adverse weather conditions properly. Pérez-Gállego et al. [23] tackle model selection for quantification tasks. Due to distribution shifts in data for quantification problems, they employ dynamic quantifier ensemble selection to select a model trained on a dataset most similar to the given test sample. To effectively select a model that precisely predicts the next sample in time series forecasting on data streams, Boulegane et al. [3] employ Multi-Target Regression (MTR). Given an ensemble of models, they assume that for temporal data streams, each model in the ensemble is an expert in some area of the stream. To select the model, they simultaneously assess each model in an ensemble based on its ability to produce a good result for the given test sample.

DILAM [18] addresses distribution shifts using incremental learning through activation matching to prevent catastrophic forgetting. They store affine transformations (scale γ and shift β) of batch normalization layers [16] in a memory bank. For each target domain, the models are adapted, and their corresponding affine transfor-

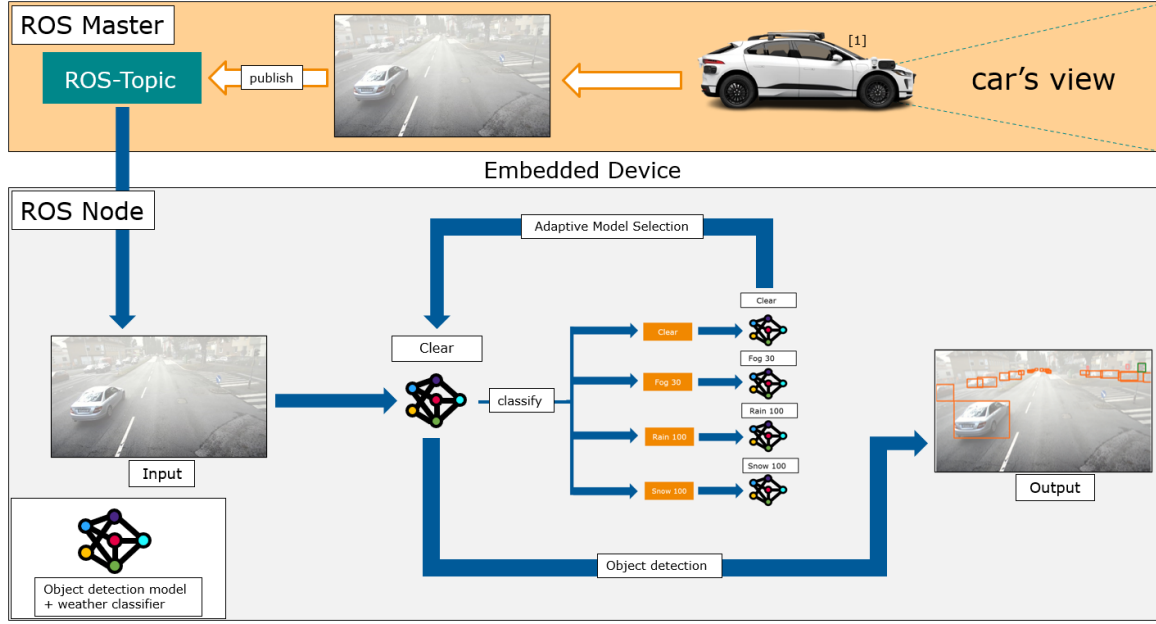


Figure 2. The Pipeline framework consists of two stages: data acquisition (top) and perception (bottom). In the data acquisition stage (ROS-master), a sensor setup composed of multiple cameras and other sensors captures road images. The second stage (ROS-node), receives the data, recognizes the current domain, and performs object detection with an expert domain model. Image marked with [1] was adapted from ¹.

mations are stored in a memory bank. During inference, their plug-and-play framework seamlessly substitutes the model’s current transformations with the pre-stored transformations specific to the target domain based on a learned domain classifier. However, changing weights during runtime as in [18] is inefficient. For embedded devices, models are converted to TensorRT, which makes changing parameters during runtime inefficient. Instead, we adopted model selection, which ensures better real-time performance and responsiveness on embedded devices.

Model compression. The need for model compression arises due to the immense network parameters in recent detection architectures. Due to memory and processing power restrictions, state-of-the-art models are unsuitable for edge devices. Reducing the model complexity and, thus, model size requires less memory to store the model. This enables us to store the model directly on the much faster on-chip memory, compared to the slow off-chip DRAM, a large storage area outside the CPU [28].

LeCun et al. [17] first show that a network can be pruned by removing weights that do not significantly affect the models’ performance. However, this approach is an unstructured pruning technique that does not consider the network structure and its layers. This method is suitable for dense layers where the weights are independent. However, this leads to problems for convolution layers where kernels share weights across spatial locations. In contrast to [17], Polyak and Wolf [24] preserve the structure of the neural network and apply channel-level prun-

ing. They either prune each layer’s input or output channels. The task is to first identify the importance of each channel by looking at the activation output variance and then filter and eliminate insignificant ones. Polyak and Wolf [24] tackle this issue by eliminating channels with the least contribution variance. Unlike unstructured pruning [17] or channel-level pruning [24], our approach simplifies pruning by pre-filtering insignificant weights using an L^1 penalty, making structured pruning more effective.

Knowledge distillation. The idea of knowledge distillation [15] is to transfer knowledge from a larger, highly accurate teacher model to a smaller, less accurate student model by computing a soft loss with the predictions of the teacher network on top of the data loss. Sau and Balasubramanian [29] extend knowledge distillation by making a student network learn from multiple teachers via logit perturbation. However, they do not directly employ multiple teacher networks but inject noise and perturbations into the teacher outputs. By doing so, they effectively simulate multiple teachers. Moreover, injecting noise into the teacher outputs introduces noise in the loss, thus creating a regularization effect. Chen et al. [6] extend the knowledge distillation workflow from [15] by considering activation responses from intermediate layers of the teacher network. This guides the student network in the correct direction and improves its accuracy. Our work draws inspiration from [6] in using knowledge distillation to improve object detection. However, instead of using intermediate layer activations, we focus on distilling knowledge through a combination of classification and bounding box loss com-

¹<https://www.roadtoautonomy.com/metaverse-waymo-spending/>.

puted with the final teacher and student outputs.

3. Method

During data recording, on-device filtering of data samples is crucial to minimize unnecessary memory consumption and processing costs for subsequent labeling or inspection. We present our object detection pipeline designed for diverse weather conditions through adaptive model selection. In addition, we provide a pruning and knowledge distillation strategy for real-time detection on embedded devices that creates highly optimized models.

3.1. Perception pipeline

Framework. With our detection pipeline, we aim to detect objects in real-time on embedded devices. It consists of various YOLOv3 detection models, each an expert for a specific weather condition (*i.e.*, *clear*, *rain* and *fog*), and a weather recognition module. After classifying the prevailing weather, the appropriate model for detection is selected and applied. We integrate our pipeline into a framework based on the Robot Operating System (ROS) [25] as illustrated in Fig. 2. It consists of two stages: data acquisition (top) and perception (bottom). In the first stage, data acquisition, a sensor setup composed of multiple cameras and optional other sensors such as LiDAR acts as the ROS master, providing sensor recordings as a data stream. The second stage is a ROS node that receives incoming data from the previous stage and runs our perception pipeline on embedded devices like the Jetson AGX Orin, Jetson Nano, or Raspberry PI 4.

Weather recognition. Similar to Leitner et al. [18], we reuse layers from the YOLOv3 backbone and employ a linear classification head to recognize the weather conditions. Therefore, we reduce the additional overhead from a separate model to a tiny linear layer. Furthermore, to get a model that performs one single forward pass, we integrate the weather recognition into the forward pass of the object detection. During the weather recognition model training, we only adjust the weights of the classification head and leave the rest of the network frozen.

3.2. Model compression

Embedded devices are constrained in resources and performance. Therefore, model compression is needed to reduce the network complexity, speeding up the inference. As illustrated in Fig. 3, model compression consists of multiple steps. To effectively reduce the model size, we first need to eliminate redundant weights from the model. However, the question of which weights are essential and which can be safely pruned without affecting the performance of the resulting pruned model arises. We start by looking at our base model, YOLOv3, which follows the YOLO network architecture and shares a common pattern throughout the network: a convolution layer followed by a batch normalization (BN) [16] layer. To effectively leverage this structure, we look closer at the BN layers.

Batch normalization (BN) [16] in deep neural networks improves stability while training and speeds up the convergence of the model. The BN layers first normalize the input and afterward scale and shift it to reduce internal covariance shifts [32]. The normalization process of BN layers is described as follows:

$$z = \frac{x_{in} - \mu_{x_{in}}}{\sqrt{\sigma_{x_{in}}^2 + \epsilon}}, \quad (1)$$

where x_{in} is the output of the previous convolution layers, $\mu_{x_{in}}$ the mean of x_{in} and $\sigma_{x_{in}}^2$ the variance. By normalizing the input, we will have zero mean and unit variance. This, however, decreases the representational power of the network. BN layers introduce γ and β parameters to retain the representational power. These parameters need to be learned by the network, where the β parameter learns the optimal shift for each BN layer and γ the optimal scaling factor. The output of BN Layers is described as follows:

$$z_{out} = \gamma \cdot z + \beta, \quad (2)$$

where z is the normalized data [21].

Sparse training. Inspecting Equ. 2 in detail, we can conclude that a smaller BN scale factor γ indicates less influence on the corresponding channel in the convolution layer. Hence, we aim to get a network structure where only a few key features of the network (high γ) are responsible for the final detection result while most channels have a γ close to zero [21]. However, the model should still learn a meaningful representation and provide highly accurate detections. Afterward, we can safely prune away the convolution channels along with the corresponding scale and shift factors γ and β for channels contributing minimally to the final detections.

We employ sparse training to get such a sparse network representation of the YOLOv3 model, where the detection result depends only on a small number of key features within the network. Leveraging a technique called proximal gradients, we compute the gradients w.r.t the regular YOLOv3 loss function. Afterward, we apply a proximal operator, namely, a soft-thresholded L^1 penalty on the γ factors of the BN layers. By applying soft thresholding, we encourage the γ factors to become zero or close to zero and hence induce sparsity into the network. The final loss function during sparse training is described as follows:

$$L = \sum_{(x,y)} l(f(x, \theta), y) + \lambda \sum g(\gamma), \quad (3)$$

where $l(f(x, \theta), y)$ is the YOLOv3 loss using the parameter vector θ and $g(\gamma)$ is the soft-thresholded L^1 penalty on the BN γ factors described as follows:

$$g(\gamma) = \text{sign}(\gamma) \cdot \max(|\gamma| - \tau, 0). \quad (4)$$

τ denotes the threshold. Every value below this threshold will be set to zero. The hyperparameter λ represents the balance between YOLOv3 loss and L^1 penalty [21].

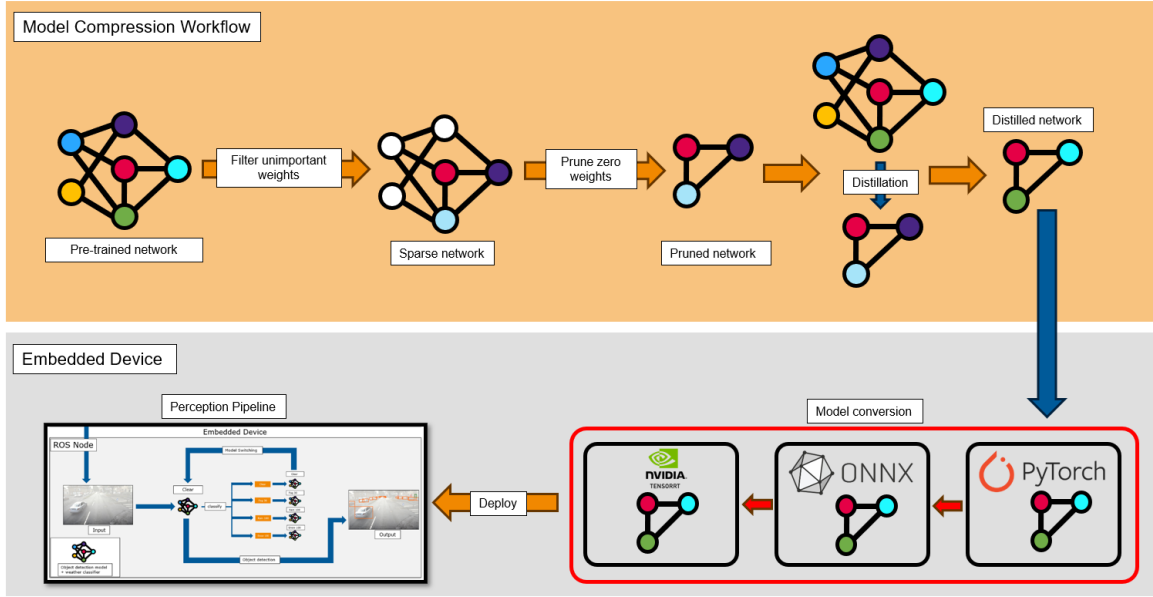


Figure 3. Overview of our method. Firstly, we sparsely train a network to filter unimportant weights. Secondly, the insignificant weights from the sparse network are pruned. Thirdly, the initial large pre-trained network distills knowledge into the pruned network. Afterward, we convert this distilled network from PyTorch to ONNX and TensorRT on the embedded device. Finally, we apply the pruned and converted model within our perception pipeline.

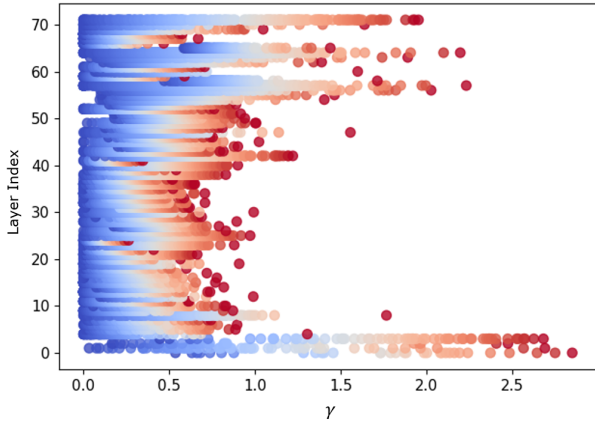


Figure 4. Distribution of BN scale factor γ after sparse training. Here we can see that not all layers have the same importance. The earlier layers have high γ values and thus are more important than the middle layers. We can also see that nearly most of the weights of the middle layers go to 0 after sparse training and hence can be pruned away safely.

Pruning. In the context of neural networks, pruning describes a technique used to eliminate weights from a network based on their importance selectively. After sparse training, we can start pruning insignificant weights from the network. The result is a model reduced in size, which also increases the inference speed. However, not all layers of the YOLO network have the same importance as shown in Fig. 4. Hence, it is crucial to prune weights also depending on the importance of layers. Following Chu et al. [7], we flatten all BN γ factor vectors in the network, concatenate them into one vector, and set a pruning per-

centage to determine a threshold. Every value below this threshold will be set to zero. Afterward, we remove every zero weight in bias, scale γ , and shift β vectors. In a subsequent step, we create a mask of the same length as each layer’s corresponding γ vector. This mask has the value one if there is a non-zero entry at the corresponding γ vector index and zero if there is a zero entry. We can safely prune convolution layers with this mask by removing each convolution channel, where the corresponding mask entry is zero [7].

Knowledge distillation. After reducing the complexity of the model and pruning weights that we have deemed insignificant in previous sections, the model’s performance may deteriorate compared to the larger base version of the network. The reason is that we may have pruned too much of the network or that weights we had deemed insignificant since they were relatively small were, in fact, significant. To counter that, we use a technique called knowledge distillation [6, 15]. Knowledge distillation describes a technique in Machine Learning where a minor student network is taught by a more extensive teacher network how to perform a specific task. In our case, the small student network is the pruned version of the base network, and the teacher network is the base version. We sample from the training dataset and run this sample through both the teacher and the student network. In the first step, we use the student network predictions and ground truth labels to compute the YOLOv3 loss. The second step is to compute the loss with the predictions of the teacher network. We first transform the teacher predictions into soft labels using the softmax function and a temperature pa-

parameter T , which controls the smoothness of the output distribution [5, 15]. The computation of the soft labels is described as follows:

$$t_{out} = \sigma \left(\frac{e^{z_i/T}}{\sum_{j=1}^N e^{z_j/T}} \right)_i, \quad (5)$$

where z_i denotes the i^{th} output of the teacher network and σ the softmax activation function. Using these soft labels t_{out} , we compute Kullback-Leibler (KL) divergence with the student class predictions s_{out} as follows:

$$KL_{loss} = \frac{KL(\log(\sigma(s_{out})), \sigma(t_{out}) \cdot T^2}{batch_size}, \quad (6)$$

where

$$KL(P||Q) = \int_{-\infty}^{\infty} p(x) \cdot \log \left(\frac{p(x)}{q(x)} \right) dx. \quad (7)$$

We use the teacher output in the loss calculation because the output of the teacher network carries significant information about relations and similarities of the predicted output. Objects similar to the actual label will have high probabilities. For instance, we expect a detector trained on a dataset with three classes, *i.e.*, *car*, *truck*, *pedestrian*, to have class label predictions for a car be close to that of a small truck but far apart from pedestrians. This example demonstrates a semantic relation between cars and trucks, which can not be incorporated with only the ground truth labels for loss calculation. Furthermore, to also consider the bounding box error, we compute a box loss with teacher-predicted bounding boxes and student bounding boxes as follows:

$$box_{loss} = \frac{1}{N} \sum_{i=1}^N (BS_i - BT_i)^2, \quad (8)$$

where BS_i is the i^{th} student bounding box and BT_i the i^{th} teacher bounding box.

Model conversion. It is insufficient to only reduce the model complexity to deploy object detection models onto embedded devices. Therefore, we need to change their structure to use the acceleration provided by embedded devices efficiently. As illustrated in Fig. 3, after successfully compressing the models, the next step is to optimize them for use on the embedded devices. In this paper, we achieve this by converting the models into the TensorRT [9] format that automatically derives essential information on how to use the underlying GPUs efficiently or accelerate inference times by restructuring the model.

4. Experiments

The experiments are split into two parts: the performance of the models tested on a workstation with high-end GPU and the performance of the models on embedded devices.

First, we provide model performances in terms of precision and model size. Afterward, we investigate the inference speed of our models measured in FPS on various embedded devices.

4.1. Dataset

In this paper, we conduct experiments on the KITTI-Dataset [11]. This dataset provides a comprehensive resource for developing and evaluating autonomous driving systems. The KITTI dataset is recorded in sunny weather conditions. However, we need additional data alongside the sunny dataset to conduct experiments for adverse weather conditions. Mai et al. [22] add artificial fog and Halder et al. [14] add artificial rain to the KITTI-clear images. KITTI-rain consists of eight severities of rain, and KITTI-fog consists of seven different severities of fog.

4.2. Implementation details

For our initial training of the YOLOv3 network, we use a batch size of 32 and a learning rate of 0.0001. We train the network for 500 epochs in total. For the sparse training, we set $\alpha=0.01$, which controls the step size for our proximal gradient, and $sr=0.001$, which controls the sparsity level. In the knowledge distillation setup, we use a batch size of 8 and train for 2000 epochs to give the teacher network enough time to teach the student network.

All models and training scripts are implemented in Python 3.8 and PyTorch 2.0.1. We train and test our models on an NVIDIA RTX 4090 GPU. For experiments on embedded devices, we run the models on three different platforms: NVIDIA Jetson AGX Orin, NVIDIA Jetson Nano, and Raspberry PI 4 Model B.

4.3. Baselines

The first step for our experiments is to get baseline models for validating the performance of our pruned and distilled models. Table 1 shows the results for our models trained on their respective domain, *e.g.*, clear, fog 30m, fog 50m, and rain 200mm/h. From these experiments, we can see that we achieve a mean average precision of 96.24 at an IoU threshold of 0.5 ($mAP@.5$) for a large float32 precision model (large-32) trained on the clear domain. Furthermore, we observe that the performance of our model for the most challenging weather condition, fog 30m, is sufficiently good, with a $mAP@.5$ of 93.36.

We can see that the model's mAP does not decrease significantly when we quantize the weights from float32 precision to float16. Additionally, we can see that the tiny models denoted as tiny-32 and tiny-16 (for float32 and float16 precision) have significantly lower mAP than our large YOLOv3 model. This is because these tiny models are 86% smaller than their base variants. Considering the model's small size, its performance is remarkable.

4.4. Performance evaluation

Before deploying our models on the NVIDIA Jetson AGX Orin, NVIDIA Jetson Nano, and the Raspberry PI 4

Model	KITTI-clear	KITTI-fog 30m	KITTI-fog 50m	KITTI-rain 200mm/h	model size
large-32	96.24	93.36	94.64	95.72	246.70 MB
large-16	96.26	93.32	94.78	95.81	123.35 MB
tiny-32	70.68	57.58	60.32	67.00	34.8 MB
tiny-16	70.70	57.55	60.26	66.93	17.8 MB

Table 1. Mean average precision at an IoU threshold of 0.5 ($mAP@.5$), when testing models on their respective domains.

Model B, we need to make sure our model’s performance measured in Average Precision (AP) and mean Average Precision (mAP) is satisfactory.

Pruned Models. Firstly, we need to mention that sparse training before pruning is absolutely crucial. The reason is that before sparse training unimportant weights have not been identified, therefore leading the pruning process to eliminate weights that are crucial to do object detection. During our experiments, we could not recover lost precision during pruning if we did not perform sparse training before pruning.

To show the efficiency of our pruning pipeline, which keeps the structure of the network intact, and our knowledge distillation pipeline, which recovers lost precision during pruning, we prune our large-32 and large-16 baseline models with different percentages, *i.e.*, 30%, 50%, and 70%. Table 2 shows the $AP@.5$ and the $mAP@.5$ for models pruned with different percentages and tested on the KITTI-clear weather domain. We can see that the Pruned models, denoted as pruned-32-30 and pruned-16-30, perform well compared to our baseline models named large-32 and large-16. Both models perform pretty well on the clear domain with a $mAP@.5$ of 93.79 and 93.78. Even if we prune 50% of the network weights, the AP does not decrease drastically.

In Fig. 4 we can see that the earlier and later layers in the YOLO network are the most important for the object detection task after the sparse training. By pruning 70% of the network, we can observe a significant drop in AP . This observation indicates that we have already eliminated significant weights from the earlier and later layers. When pruning 90% of the weights, our model degenerates and cannot detect objects anymore.

To increase the performance of the tiny models (tiny-32 and tiny-16) we apply our proposed knowledge distillation pipeline to boost its ability to detect objects. By doing so, we increase the mAP of the YOLOv3-tiny models by around 12 points, from 70.70 to 82.48. When comparing this tiny model to our large-32 baseline, we can see that we only have a 14-point difference in $mAP@.5$. This precision is particularly good considering that the tiny model is 86% smaller than the large-32 model.

4.5. Performance on embedded devices

After verifying our models’ results on a regular PC and reducing the model’s complexity, we deploy them

onto resource- and performance-constrained embedded devices. In this case, we are mainly interested in the inference speed of our models when tested on the KITTI dataset. Table 3 shows our results. We can see that for the large models (246.7 MB), we achieve 93.45 FPS when running inference on images of size 416×416 on the Jetson AGX Orin. The FPS drops considerably when we deploy the same model onto a smaller embedded device like the Jetson Nano or Raspberry PI 4. Due to the compact size and performance constraints of both of these smaller embedded devices, the performance drops by 97.52% and 99.78%, respectively. To reach the domain of real-time inference, we need to achieve at least 24 FPS to outperform the sensor recording frequency. Pruning 30% of the network weights increases our FPS from 93.45 to 100.00 on the Jetson Orin. We are more than twice as fast on the Jetson Nano and the Raspberry PI 4.

To finally reach real-time inference even on the Jetson Nano, we utilize the YOLOv3-tiny model. This model has a significantly smaller network architecture. We can see that this model is 86% smaller than our large-32 model. With this model, we achieve 23.8 FPS, and by quantizing the model’s weights to float16, we achieve 31.25 FPS. On the Raspberry PI 4, we increased the performance by more than 9 times compared to the large-32 model by using the tiny model.

Furthermore, we can also see that for the Raspberry PI 4, the float32 precision models are faster than the float16 precision models even though the latter are smaller. This is because The Raspberry has no GPU and thus does not provide GPU acceleration. Furthermore, CPUs have native support for float32 and can, therefore, handle them more efficiently than float16. Reducing the complexity of our models leads to a reduction in the number of floating point operations (FLOPS).

5. Limitations and future work

Our object detection and weather classification model is limited by how many domains it can effectively work with. We have trained both the object detection part and the weather classification part with synthetic data. Therefore, we will encounter a significant drop in precision for any domain unavailable at training time. To ensure the seamless operation of the perception system under a broader range of adverse weather conditions, including rain and snow, it is crucial to have large and diverse publicly available datasets. These datasets will support the

Model	Tested on KITTI-clear								mAP@.5	Model size
	Car	Van	Truck	Ped	Psit	Cyc	Tram	Misc		
large-32	98.26	98.81	99.28	90.53	90.38	96.30	98.97	97.37	96.24	246.70 MB
large-16	98.26	98.79	99.27	90.60	90.38	96.54	98.93	97.35	96.26	123.35 MB
pruned-32-30	97.84	97.91	98.91	86.11	80.60	93.47	99.03	96.46	93.79	173.40 MB
pruned-16-30	97.82	97.93	98.91	86.14	80.62	93.36	99.03	96.45	93.78	86.70 MB
pruned-32-50	97.22	97.41	98.35	84.80	86.20	91.64	98.40	94.60	93.56	134.00 MB
pruned-16-50	97.20	97.40	98.34	84.61	86.23	91.51	98.41	94.60	93.52	67.00 MB
pruned-32-70	93.28	91.97	93.35	70.61	60.94	77.88	87.32	75.82	81.40	76.30 MB
pruned-16-70	93.24	91.93	93.36	70.39	61.28	77.76	87.20	76.06	81.40	38.15 MB
tiny-32	86.32	74.83	81.97	59.96	59.46	65.15	80.83	56.95	70.68	34.8 MB
tiny-16	86.41	75.07	82.09	60.20	59.00	64.89	80.82	57.12	70.70	17.8 MB
tiny-32-kd	92.71	90.14	93.51	68.66	64.18	78.68	90.87	81.10	82.48	34.8 MB
tiny-16-kd	92.67	90.02	93.55	68.79	66.03	78.61	90.88	81.02	82.70	17.8 MB

Table 2. Average Precision (AP) and mean Average Precision ($mAP@.5$) for all KITTI classes using models with different pruning percentages.

Model	Tested on Architecture			Model size	GFLOPS
	Jetson AGX Orin	Jetson Nano	Raspberry PI 4		
large-32	93.45 / 80.65	2.32 / 2.04	0.32 / 0.21	246.70 MB	32.75 / 49.61
large-16	111.11 / 103.09	3.90 / 3.26	0.26 / 0.19	124.30 MB	32.75 / 49.61
pruned-32-30	100.00 / 90.90	5.55 / 4.78	0.61 / 0.49	173.40 MB	14.10 / 21.36
pruned-16-30	116.27 / 108.70	8.69 / 7.29	0.43 / 0.41	86.50 MB	14.10 / 21.36
pruned-32-50	103.09 / 94.33	7.09 / 5.99	0.83 / 0.57	134.00 MB	10.36 / 15.69
pruned-16-50	120.04 / 117.64	10.75 / 9.09	0.72 / 0.48	67.00 MB	10.36 / 15.69
pruned-32-70	111.11 / 100.00	9.90 / 8.00	1.31 / 0.84	76.30 MB	6.31 / 9.55
pruned-16-70	149.25 / 125.03	15.38 / 11.76	1.08 / 0.75	38.15 MB	6.31 / 9.55
tiny-32	181.82 / 142.85	23.80 / 20.19	3.09 / 2.32	34.80 MB	2.75 / 4.15
tiny-16	212.76 / 176.42	31.25 / 27.89	3.04 / 1.53	17.80 MB	2.75 / 4.15

Table 3. Average FPS and model size on different architectures. The values in the Model column consist of the model name, the precision, and the percentage of weights pruned. The values in the Tested on Architecture column represent the FPS. The first value denotes the FPS when running inference on images of 416×416 pixels; the second value represents the same but with images of 512×512 pixels. The FPS values are averaged over 3781 (test set) iterations. The values in the GFLOPs column represent the Giga Floating Point Operations per Second. Again, the first value denotes inference using an image of size 416×416 and the second represents inference using an image of size 512×512 .

advancement of unsupervised domain adaptation by enabling improvements in robust domain recognition and the refinement of object detection capabilities.

6. Conclusion

In this paper we studied real-time object detection in diverse weather conditions through adaptive model selection on embedded devices. We particularly focused on gaining a deeper understanding of model compression techniques to reduce model complexity and enable real-time applications on performance- and resource-constrained embedded devices. The key findings entail:

- Filtering insignificant network weights is essential to reduce precision loss during pruning and to make the model rely on key features for object detection only.
- Knowledge distillation is a suitable technique to regain the lost precision after pruning.
- Our proposed perception pipeline ensures real-time ob-

ject detection on embedded devices. It recognizes known domains, selects a suitable model, and performs robust real-time object detection without interruptions.

These findings emphasize the significance of a structured approach that reduces model size to increase inference speed on embedded devices. This enables automatic driving applications to run robustly in real time and adapt to adverse weather conditions, such as fog or rain.

Acknowledgements

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged. Furthermore, we gratefully acknowledged the financial support of the Austrian Federal Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology, the Austrian Federal Ministry of Digital and Economic Affairs, in the frame

of the Important Project of Common European Interest (IPCEI) on Microelectronics and Communication Technologies (ME/CT) implemented by austria wirtschaftsservice (aws) and the Austrian Research Promotion Agency (FFG).

References

- [1] Pedro Azevedo and Vítor Santos. Yolo-based object detection and tracking for autonomous vehicles using edge devices. In *Proc. ROBOT*, 2022. 2
- [2] Juan Borrego-Carazo, David Castells-Rufas, Ernesto Biempica, and Jordi Carrabina. Resource-constrained machine learning for ADAS: A systematic review. *IEEE Access*, 8:40573–40598, 2020. 1
- [3] Dihia Boulegane, Albert Bifet, Haytham Elghazel, and Giyyarpuram Madhusudan. Streaming time series forecasting using multi-target regression with dynamic ensemble selection. In *IEEE BigData*, 2020. 2
- [4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. ECCV*, 2020. 2
- [5] Alexandros Chariton. Knowledge Distillation Tutorial. https://pytorch.org/tutorials/beginner/knowledge_distillation_tutorial.html, 2024. Online; accessed March 14, 2024. 6
- [6] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Proc. NeurIPS*, 2017. 3, 5
- [7] Yun Chu, Pu Li, Yong Bai, Zhuhua Hu, Yongqing Chen, and Jiafeng Lu. Group channel pruning and spatial attention distilling for object detection. *AI*, 52(14):16246–16264, 2022. 5
- [8] European Commission. Annual accident report. https://road-safety.transport.ec.europa.eu/european-road-safety-observatory/statistics-and-analysis-archive/annual-accident-report_en, 2021. Online; accessed May 23, 2024. 1
- [9] NVIDIA Developer. NVIDIA TensorRT. <https://developer.nvidia.com/tensorrt>, 2024. Online; accessed April 15, 2024. 6
- [10] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proc. ICCV*, 2019. 2
- [11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? the KITTI Vision Benchmark Suite. In *Proc. CVPR*, 2012. 6
- [12] Ross Girshick. Fast r-cnn. In *Proc. ICCV*, 2015. 2
- [13] Brent Griffin. Mobile robot manipulation using pure object detection. In *Proc. WACV*, 2023. 2
- [14] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-Based Rendering for Improving Robustness to Rain. In *Proc. ICCV*, 2019. 6
- [15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, abs/1503.02531, 2015. 2, 3, 5, 6
- [16] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. ICML*, 2015. 2, 4
- [17] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Proc. NeurIPS*, 1989. 3
- [18] Stefan Leitner, M Jehanzeb Mirza, Wei Lin, Jakub Mi-corek, Marc Masana, Mateusz Kozinski, Horst Possegger, and Horst Bischof. Sit Back and Relax: Learning to Drive Incrementally in All Weather Conditions. In *Proc. IV*, 2023. 2, 3, 4
- [19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. ICCV*, 2017. 2
- [20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proc. ECCV*. Springer, 2016. 2
- [21] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proc. ICCV*, 2017. 4
- [22] Nguyen Anh Minh Mai, Pierre Duthon, Louahdi Khoudour, Alain Crouzil, and Sergio A. Velastin. 3D Object Detection with SLS-Fusion Network in Foggy Weather Conditions. *Sensors*, 21(20), 2021. 6
- [23] Pablo Pérez-Gállego, Alberto Castaño, José Ramón Quevedo, and Juan José del Coz. Dynamic ensemble selection for quantification tasks. *IF*, 45:1–15, 2019. 2
- [24] Adam Polyak and Lior Wolf. Channel-level acceleration of deep face representations. *IEEE Access*, 3:2163–2175, 2015. 3
- [25] Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote, Jeremy Leibs, Eric Berger, Rob Wheeler, and Andrew Ng. ROS: an open-source Robot Operating System. In *Proc. ICRA*, 2009. 4
- [26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, 2016. 2
- [27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NeurIPS*, 2015. 2
- [28] Babak Rokh, Ali Azarpeyvand, and Alireza Khanteymoori. A Comprehensive Survey on Model Quantization for Deep Neural Networks in Image Classification. *ACM TIST*, 14(6), 2023. 3
- [29] Bharat Bhusan Sau and Vineeth N Balasubramanian. Deep model compression: Distilling knowledge from noisy teachers. *arXiv*, abs/1610.09650, 2016. 3
- [30] Ruohuai Sun, Chengdong Wu, Xue Zhao, Bin Zhao, and Yang Jiang. Object recognition and grasping for collaborative robots based on vision. *Sensors*, 24(1):195, 2023. 2
- [31] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proc. CVPR*, 2020. 2
- [32] Eric Wong. Distribution shift. https://riceric22.github.io/assets/debugml/distribution_shift.pdf, 2022. Online; accessed April 27, 2024. 4
- [33] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *IEEE*, 111(3):257–276, 2023. 2