Appendix - Mind the Gap: Transferring Labels to Align Object Detection Datasets

Mikhail Kennerley[1,2] Angelica I. Aviles-Rivero[3] Carola-Bibiane Schönlieb[4] Robby T. Tan[1,5]

[1]National University of Singapore [2]Institute for Infocomm Research (A*STAR) [3]Tsinghua University [4]University of Cambridge
[5]ASUS Intelligent Cloud Services *

mikhailk@u.nus.edu alives@tsinghua.edu.cn c.b.schoenlieb@damtp.cam.ac.uk
robby.tan@nus.edu.sg

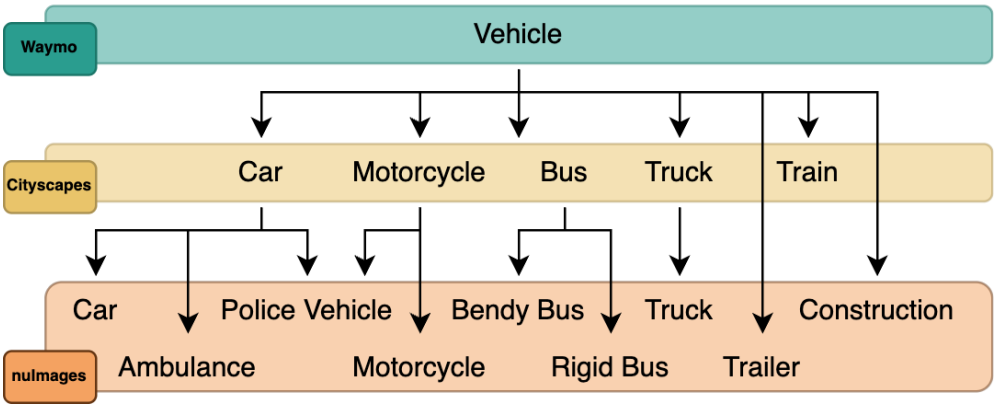

Figure 1. Illustration of class granularity divergence for vehicle-related labels across Waymo, Cityscapes, and nuImages. Waymo uses broad classes, while nuImages is the most fine-grained.

1. Additional Discussion

1.1. Scaling Label-Aligned Transfer

LAT adopts a modular multi-stage pipeline designed for flexible dataset integration. In the first stage, an object detector is trained independently for each of the N datasets, scaling linearly with N and enabling parallel training. In the second stage, pseudo-labels are generated by running each model on the remaining N-1 datasets, resulting in N(N-1) inference passes. While this introduces quadratic scaling in pre-processing, the computation is parallel and easily distributed.

This design provides two key benefits: (1) it preserves dataset-specific annotation conventions before unification. (2) it allows components to be updated independently without retraining the entire system. We view this overhead as a strategic trade-off for high-fidelity label transfer, particularly suited for scenarios involving an intermediate number of datasets. While scaling beyond this range may pose challenges, these are confined to the training phase. A key advantage of LAT is that it introduces no additional overhead at inference time, once trained, the downstream detector operates with standard runtime efficiency.

2. Additional Experimental Details & Experiments

2.1. Additional Details on Benchmarks

Cityscapes ↔ nuImages ↔ Waymo. To complement the main paper, Figure 1 visualizes how the three datasets diverge in label granularity, using the vehicle super-class as an example. Waymo collapses multiple fine-grained categories into a single *vehicle* label, while nuImages introduces highly specific subclasses, and Cityscapes adopts an intermediate taxonomy. This structural mismatch illustrates why semantic alignment across these datasets is non-trivial. Following the main setup, we use 3,000 images per dataset for controlled evaluation.

Method	Model	Method	Dataset		
			Cityscapes	nuImages	Waymo
Baseline	YOLO	-	53.9	37.1	45.6
LAT	YOLO	Label Transfer	**59.1**	**37.6**	**47.2**

Table 1. YOLO-based downstream detectors in class-divergent setting. LAT consistently improves performance even with a single-stage detector like YOLO.

Method	Model	Method	Dataset			
			Cityscapes	ACDC	BDD100K	SHIFT
Baseline	YOLO	-	53.9	41.4	56.0	**64.2**
LAT	YOLO	Label Transfer	**57.9**	**42.1**	50.1	60.1
LAT (Long Train)	YOLO	Label Transfer	58.2	43.7	**56.4**	**64.2**

Table 2. YOLO-based downstream detectors in small–large dataset setting. ACDC sees large gains under LAT, replicating trends seen with FRCNN and RT-DETR.

2.2. Experimental Set-up

We implement LAT using the FRCNN [6] framework built on Detectron2 [8]. DINOv2 [5] is employed as a frozen feature extractor with pre-trained weights. In our PPG module, random jittering and ground-truth label removal are applied at rates of 0.5 and 0.05, respectively. LAT is trained for 30,000 iterations using a learning rate of 0.2 and a batch size of 4 on a single RTX 3090 GPU.

For downstream training, we use a FRCNN model with a modified weighted cross-entropy loss, where the weight is derived from the confidence score of the pseudo-label. This model, as well as the initial pseudo-label generation model, is trained for 50,000 iterations with a fixed learning rate of 0.2 and a batch size of 16. In addition, we train RT-DETR and YOLOv11 models as our downstream detector for comparisons to more modern detectors as compared to FRCNN. These detectors are trained for 300,000 iterations with a batch size of 64. AdamW is used as the optimizer with a learning rate of 0.001 and momentum of 0.9. All downstream models are trained using four NVIDIA RTX 3090 GPUs.

2.3. results

Consistent performance on YOLOv11. We conduct additional experiments using YOLOv11 [3] to verify that the performance gains from LAT's refined pseudo-labels are not tied to a specific model architecture. As shown in Table 1 and Figure 2, LAT-trained labels consistently improve performance on YOLOv11, complementing the results already demonstrated with FRCNN and RT-DETR in the main paper. This confirms that the benefits of LAT generalise across detectors, demonstrating its effectiveness as a model-agnostic label transfer framework.

Performance on simpler transfer protocols. We compare our method to LGPL [4] in Table 3, using the synthetic-to-real transfer setting from Synscapes [7] to Cityscapes [2]. We consider this a simpler transfer scenario, as both datasets share identical class labels and exhibit similar semantic structures. Moreover, the setup involves a one-to-one label space transfer, reducing the need for LAT's full design capabilities, such as many-to-one label alignment and the performance gains that emerge from integrating multiple source datasets. Nevertheless, LAT matches the performance of the state-of-the-art LGPL method, demonstrating its effectiveness even under minimal transfer complexity. We note that LGPL results are reported directly from the original paper using mAP@[.5:.95] as a metric, as public code was not available at the time of writing.

Qualitative results: LAT mitigates pseudo-label noise. We illustrate LAT's effectiveness in addressing pseudo-label noise in Figures 3,4, and5. Each figure presents three columns: the first shows initial pseudo-labels from the upstream detector in the target label space; the second shows LAT-refined pseudo-labels in the same label space; and the third displays the ground-truth annotations in the original source label space. Note that class names may differ between the pseudo-labels and ground-truth columns due to label space discrepancies.

Class-aware context is critical for robust aggregation. Table 4 ablates the aggregation module used in SFF. Removing SFF entirely leads to the largest drop, reducing performance from 60.1 AP to 57.9 AP, confirming that cross-proposal aggregation is a key component of LAT. Among the two branches, the visual-context (VC) path contributes

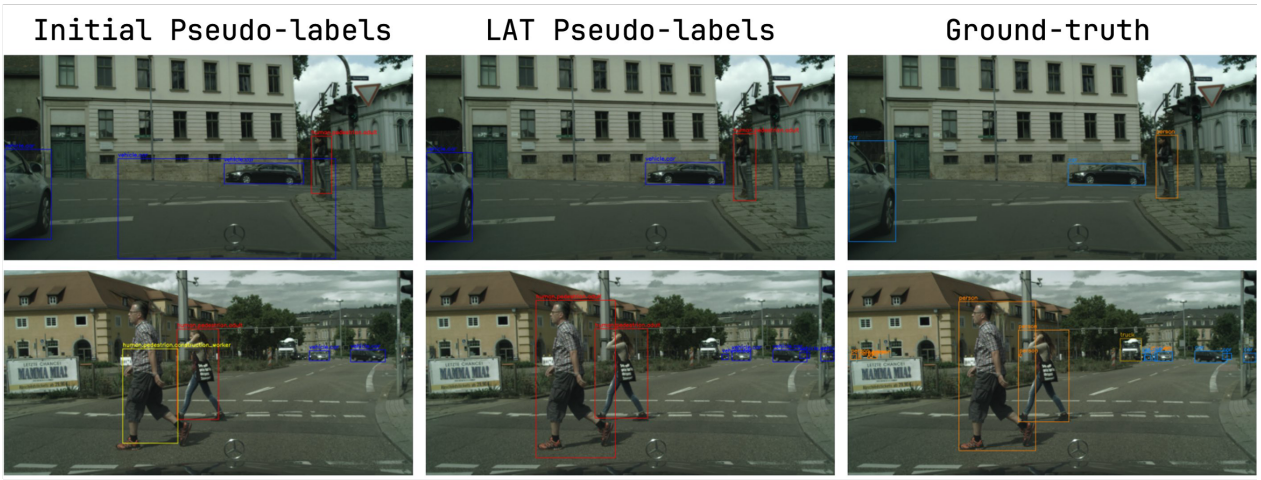

Figure 2. LAT is able to identify and remove false pseudo-labels (top row) and correct the classification and bounding box of an object (bottom row) from inaccurate pseudo-labels.

Model	Def-DETR	Faster RCNN
Baseline	32.9	38.7
Pseudo-Label	30.7	36.9
Pseudo-Label + Filtering	33.0	39.1
LGPL [4]	34.5	**39.7**
LAT	**34.7**	39.6

Table 3. Performance on Synscapes → Cityscapes (1:1 transfer). LAT matches the performance of state-of-the-art LGPL despite the simpler one-to-one transfer setup.

	CITYSCAPES
Full Model	**60.1**
No SC Weighting	59.8
VC Only	59.5
VC Pre-Logit	59.0
VF Only	58.2
No Attention	57.5

Table 4. Ablation of SFF on Cityscapes. Removing SFF entirely causes the largest drop, confirming the importance of cross-proposal aggregation. The visual-context (VC) branch is more effective than the visual-feature (VF) branch, and using class scores as context performs better than using pre-logit features, supporting the value of class-aware semantic context.

more strongly than the visual-feature (VF) path: VC Only achieves 59.5 AP, while VF Only reaches 58.2 AP. This suggests that class-informed contextual transfer is more effective than feature-only self-attention for aligning heterogeneous label spaces. Replacing class-score context with pre-logit features also lowers performance, from 60.1 AP to 59.0 AP, indicating that semantically explicit class-score signals are more useful than generic feature-space context. Finally, removing score-confidence weighting causes a smaller drop to 59.8 AP, showing that confidence-aware gating provides an additional robustness benefit. Overall, the best performance is obtained when both branches, class-score context, and score-confidence weighting are used together.

Label Transfer improves robustness to noisy supervision. We observe that detectors trained on LAT-generated labels consistently outperform those trained on standard pseudo-labels. This highlighting LAT's ability to mitigate noise introduced during pseudo-label generation. This robustness stems from LAT's integration of ground-truth annotations and multi-source pseudo-labels, allowing the model to resolve both semantic and spatial inconsistencies. Figure 2 demonstrates this and we provide additional qualitative examples where LAT corrects various forms of pseudo-label noise, including missing annotations, misclassified or misaligned boxes, and false positives in the supplementary materials.

TIDE [1] shows that pseudo-label supervision improves recall, while LAT makes it more reliable. Compared with Target Only training, both Pseudo-labels and LAT substantially reduce omission errors on Cityscapes, with *Miss* dropping from 24.66 to 15.73 and 15.80, and *FalseNeg* decreasing from 36.22 to 28.56 and 28.21, respectively. This indicates that transferred supervision mainly improves recall by recovering previously missed objects. At the same time, LAT produces more reliable supervision than standard Pseudo-labels, reducing *Cls* from 6.39 to 5.40 dAP, *Bkg* from 1.80 to 1.65 dAP, *Both* from 1.45 to 1.25 dAP, and

Error Type	Target Only	Pseudo-labels	LAT
Miss	24.66	**15.73**	15.80
Loc	**5.17**	6.78	6.83
Cls	5.68	6.39	**5.40**
Bkg	**1.24**	1.80	1.65
Both	**1.09**	1.45	1.25
Dupe	**0.01**	0.06	0.05
FalseNeg	36.22	28.56	**28.21**
FalsePos	**7.04**	9.56	8.79

Table 5. TIDE error analysis on Cityscapes. Values are dAP, where lower is better. Pseudo-label supervision methods strongly reduces recall-related errors compared with Target Only training, while LAT preserves these gains and achieves the lowest classification and false negative errors overall.

FalsePos from 9.56 to 8.79 dAP, while maintaining similar localization error. Overall, TIDE suggests that pseudo-labeling provides the main recall gain, and LAT improves its reliability by suppressing part of the noise introduced by naive pseudo-labels.

References

[1] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. In *European Conference on Computer Vision*, pages 558–573. Springer, 2020. 3

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2

[3] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO, 2023. 2

[4] Yuan-Hong Liao, David Acuna, Rafid Mahmood, James Lucas, Viraj Uday Prabhu, and Sanja Fidler. Transferring labels to solve annotation mismatches across object detection datasets. In *The Twelfth international conference on learning representations*, 2024. 2, 3

[5] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. 2015. 2

[7] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing, 2018. 2

[8] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 2

Figure 3. Qualitative results of Cityscapes target label space in class-divergent setting with nuImages dataset. **Row 1:** LAT recovers truck instances missing from initial pseudo-labels. **Row 2:** LAT refines truck bounding boxes and detects small objects. **Row 3:** LAT recovers heavily obscured cars. **Row 4:** LAT detects pedestrians omitted by upstream pseudo-labels. **Row 5:** LAT identifies foreground and background trucks. **Row 6:** LAT recovers vehicles under adverse weather conditions (rain).

Figure 4. Qualitative results of nuImages target label space in class-divergent setting with Cityscapes dataset. **Row 1:** LAT detects an occluded pedestrian missed by initial pseudo-labels. **Row 2:** LAT corrects noisy human-annotated ground-truth. **Row 3:** LAT removes an erroneously predicted car from initial pseudo-labels. **Row 4:** LAT recovers a bus instance omitted by the upstream model. **Row 5:** LAT refines pedestrian bounding box and corrects its class label. **Row 6:** LAT recovers small-scale pedestrian and bicycle instances.

Figure 5. Qualitative results of nuImages target label space in class-divergent setting with Waymo dataset. **Row 1:** LAT detects background pedestrians and vehicles under adverse nighttime conditions. **Row 2:** LAT detects both foreground pedestrians and background vehicles at night. **Row 3:** LAT recovers multiple vehicles in rainy nighttime scenes. **Row 4:** LAT detects vehicles and pedestrians in rainy conditions. **Row 5:** LAT recovers multiple vehicle instances in rain. **Row 6:** LAT detects a pedestrian in an uncommon pose.