Bachelor Thesis
Supervisor: Kirillova, Nadezda

Multimodal Domain Adaptation for enhancing Airplane Detection and Tracking

Adapting FairMOT with Pseudo-Labeling via Vision-Language Models

Maximilian Wawrina

Institute of Visual Computing
Graz University of Technology, Austria

Graz, March 24, 2026

Abstract

This thesis investigates multimodal domain adaptation for airplane multi-object tracking (MOT) in airport surveillance footage. In practice, MOT models often degrade when deployed in a new domain due to changes in viewpoint, background scenery and environmental conditions. This is challenging to address due to the difficulty of acquiring high-quality labels for target domains. To mitigate this domain shift, a baseline airplane tracker based on FairMOT was first fine-tuned on a public airport surveillance dataset, and subsequently applied to a new, unseen domain. To enable domain adaptation without requiring manual labels in the new domain, a pseudo-labeling pipeline was introduced. Low-confidence detections were evaluated using a vision-language model (VLM), which was selected through a custom airplane classification benchmark. The refined detections were used as pseudo-labels to retrain the tracker on the target domain, thereby performing domain adaptation through self-training. The adapted model is compared to the baseline to assess the impact of VLM-assisted pseudo-labeling. Results show that incorporating vision-language signals for pseudo-label selection can improve cross-domain tracking performance, although the effectiveness is sensitive to prompting strategies. Overall the findings demonstrate that VLM-guided pseudo-labeling combined with subsequent retraining provides an effective mechanism for domain adaptation in airplane multi-object tracking.

Keywords: *Bachelor Thesis, Domain Adaptation, Vision Language Models, FairMOT, multi-object detection and tracking, airplane detection and tracking, InternVL2, CLIP, SigLIP*

Table of Contents

1 Introduction

Object detection and tracking systems often struggle to generalize beyond the specific visual conditions on which they were trained. This problem, referred to as domain adaptation [25], occurs when applying a pre-trained model on a new, unseen domain with differing FOVs, viewpoints, weather and lighting conditions, and more. Even slight shifts in these domain conditions can degrade model performance significantly due to the underlying neural network not having been exposed to such variations during training.

This issue becomes especially apparent when applying models to detect and track airplanes using airport surveillance video footage. Although a neural network can be trained on extensive data from one airport, its performance would decline significantly if utilized on differing airports, or even just different cameras of the same airport. This makes it difficult to deploy such models widely, emphasizing the need to improve their robustness under domain shifts.

Recently, vision-language models (VLMs) [4] [17] [29] have demonstrated strong generalization capabilities across varying visual domains. In this thesis, VLMs are used as a tool to support domain adaptation. The core idea is to use VLMs to validate low-confidence detections produced by a baseline tracker in an unseen domain. These refined detections are treated as pseudo-labels and are subsequently used to retrain the tracking model on the target domain. In this way, domain adaptation is performed through self-training, without requiring manual annotations in the new domain. Importantly, the proposed approach does not require access to labeled source data, but only an existing pre-trained tracking model, unlabeled target data and a VLM. Since the VLM evaluates detections independently of the tracking model, the method is architecture agnostic and requires no modifications to the underlying tracking model.

To investigate this approach, a multi-object tracking (MOT) model based on FairMOT [31] was trained to detect and track airplanes on a suitable dataset [6]. The resulting baseline model was then applied to sequences from an unseen domain using a low confidence threshold in order to obtain a large set of candidate detections. These detections were evaluated using different VLMs, which were compared through a custom airplane classification benchmark to select a suitable model. The validated detections were then used as pseudo-labels to retrain the tracker on the target domain. The performance of the adapted model is compared to the original baseline to assess the effectiveness of VLM-guided pseudo-labeling. The tracking architecture itself is based on convolutional neural networks (CNNs) rather than transformers. While transformer-based vision models often achieve higher accuracy, CNN-based pipelines remain attractive for real-time applications due to their lower computational overhead and suitability for resource-constrained environments.

The thesis is structured as follows: Section 2 provides a theoretical background on multi-object tracking frameworks, domain adaptation and vision-language models. Section 3 introduces the proposed methodology, including the problem definition and the overall experimental design for pseudo-label generation and retraining. It also describes the experimental setup with a focus on the selected dataset, pre-processing steps, and the classification and tracking experiments. Section 4 presents the results of the baseline and the adapted models, followed by a comparative analysis. Finally, Section 5 concludes the thesis and Section 6 outlines potential directions for future work.

Research Questions

This thesis investigates the following research questions:

1. How severely does the FairMOT baseline degrade when applied to an unseen domain?
2. Which VLM is most suitable for generating pseudo-labels for airplane detection in the target domain?
3. Does retraining on VLM pseudo-labels improve tracking performance compared to the FairMOT baseline model?

2 Background and Related Work

2.1 Multi-Object Tracking

In modern computer vision, multi-object tracking can be defined as the problem of estimating the trajectories of objects of interest across a sequence of frames. MOT is most commonly formulated as a tracking-by-detection framework. Objects are detected in each frame as bounding boxes and then associated over time to form continuous object trajectories. Classical multi-target tracking typically relies on data association techniques such as Multiple Hypothesis Tracking (MHT) [19] and Joint Probabilistic Data Association (JPDA) [20]. These methods are computationally expensive and memory intensive [14], which limits their suitability for online MOT. In contrast, the Simple Online and Realtime Tracking [1] framework (SORT) proposed by Bewley et al. was able to achieve a significant improvement in MOT performance by replacing the traditional methods with more efficient techniques such as the Kalman filter and the Hungarian algorithm. This shift toward simpler motion models and efficient assignment significantly improved real-time performance. Deep SORT [27] was later proposed as an improved framework which introduced appearance feature information for re-identification, which the original SORT framework lacked.

Since the introduction of SORT and Deep SORT, other methods such as FairMOT [31] were proposed, which aimed to improve and modernize online MOT by combining the previously separate, two stages of detection and re-identification into a single network. In FairMOT [31], it is noted that earlier one-shot trackers that jointly learn detection and identity embedding often underperform two-step pipelines. FairMOT aims to address this limitation by jointly learning detection and identity embedding within a single network while ensuring balanced optimization between both tasks.

Compared to SORT and Deep SORT, which rely on separate motion and appearance components, FairMOT jointly produces detections and identity embeddings within a unified architecture. This design makes it particularly suitable for studying domain shift, as both detection quality and identity consistency are affected when transferring to a new domain. Because the model outputs final tracking results directly in an online setting, it provides a clean baseline for evaluating the effect of VLM-guided pseudo-labeling and subsequent retraining.

2.2 Domain Adaptation for MOT

In computer vision, domain shift refers to changes between a source domain and a target domain caused by factors such as lighting conditions, weather, differing viewpoints or background variations. When a model trained on a source domain is deployed in a different target domain, these shifts often lead to a significant drop in performance [25].

Domain adaptation techniques address this problem by adapting a model trained on a source domain so that it performs reliably on a new target domain. Since collecting labeled data for every possible deployment scenario is costly and often impractical, domain adaptation aims to transfer knowledge from the source domain to the target domain while minimizing additional annotation effort.

Domain adaptation settings are commonly categorized based on the availability of labeled data in the target domain. In supervised settings, labeled target samples are available. In semi-supervised settings, only a small subset of target data is labeled. In unsupervised domain adaptation, no labeled target samples are available, which makes adaptation particularly challenging [25].

Several unsupervised domain adaptation methods have been proposed to improve generalization to an unlabeled target domain by leveraging labeled source data during training. Examples include approaches that explicitly reduce the feature distribution gap between source and target representations [15] [22], methods that learn domain-invariant features via a domain

discriminator [11] [23], and frameworks that attempt to improve the learned representation by reconstructing unlabeled target samples [3] [13]. While effective in many settings, these approaches typically assume access to labeled source data and often require source domain samples during the adaptation, which may be impractical once a pre-trained model is deployed or when source data cannot be shared.

In this thesis, the focus lies on the unsupervised domain adaptation setting, where no labeled samples are available in the target domain. The investigated approach leverages pseudo-labeling and self-training, which only requires a pre-trained model, unlabeled target domain data and access to a VLM, thereby avoiding the need for source data access entirely whilst also retaining the real-time tracking performance of the MOT model.

2.3 Vision-Language Models

Vision-language models learn joint representations by aligning images and natural-language descriptions in a shared embedding space. A representative example of a large-scale image-text training model is CLIP [17], which stands for Contrastive Language-Image Pre-training. In the case of CLIP, the network was trained on a dataset consisting of 400 million image-text pairs obtained from publicly available online sources.

A key advantage of VLMs is their ability to generalize beyond the domain on which they were trained. This can be traced back to their supervision being derived from natural language descriptions, which exposes the model to a much broader range of visual concepts than a fixed-label dataset can provide. An important factor behind this robustness is the ability to perform zero-shot classification. In a zero-shot setting, the model predicts classes without being trained on task-specific labeled examples for those classes. Instead, images are categorized by comparing them to natural-language descriptions. As a result, in the case of CLIP, it ”transfers non-trivially to most tasks and is often competitive with a fully supervised baseline without the need for any dataset specific training.” [17]. This makes VLMs a promising tool in settings where domain-specific labels are limited or unavailable, as they can provide additional supervision signals without requiring manual annotation.

In comparison to CLIP, SigLIP [29], which stands for Sigmoid loss for Language-Image Pre-training, proposes the use of sigmoid scoring instead of CLIP's approach of using softmax loss. This allows SigLIP to perform well even with smaller batch sizes during pre-training, improving memory efficiency while maintaining strong zero-shot classification performance. Finally, this thesis also includes an InternVL [4] model, as it differs significantly from the aforementioned networks. InternVL is a more recent VLM family that combines a large-scale vision encoder with a language model-based alignment module, enabling both contrastive and generative vision-language tasks. These three models were selected to evaluate their suitability for generating reliable pseudo-labels in the target domain. Their architectural differences and training strategies allow for a meaningful comparison in the context of cross-domain airplane tracking.

3 Methodology

3.1 Overview of the Method

In this work, the performance degradation of a FairMOT-based airplane tracker is addressed when it is deployed in a visual domain that differs significantly from the training data. In addition to the issue of domain shift, the FairMOT framework must also be adapted to enable the tracking of airplanes, since the original model was created for pedestrian detection and tracking. In the course of this thesis, it becomes clear that applying a standard confidence threshold is ill-suited when the FairMOT baseline tracker is transferred to a new, unseen domain. Under domain shift, many true airplane instances receive reduced detector confidence scores and would

therefore be missed if only standard-confidence detections were retained. This is problematic because missing instances of airplanes prevents stable tracking, since the network cannot assign IDs to objects that were never detected. To address this problem, the proposed method is divided into three distinct phases: baseline FairMOT pre-training, VLM benchmarking for pseudo-label generation and VLM-guided self-training for target domain adaptation.

The first step of this methodology is the pre-training of the FairMOT baseline model on a suitable dataset. This pipeline begins by splitting the dataset into a source domain used for training and a target domain that is withheld for the subsequent adaptation experiment. The source domain is further split into training, validation and test sets, on which the FairMOT model with a DLA-34 [28] backbone is trained. This stage results in a FairMOT baseline model which is capable of detecting and tracking airplanes, despite the fact that FairMOT was originally designed for detecting and tracking pedestrians. This proposed course of action is visualized in figure 1.

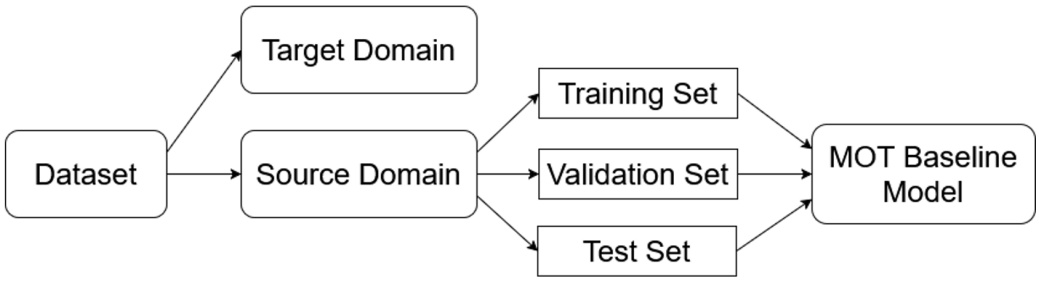


Figure 1: Overview of the proposed FairMOT baseline pre-training pipeline

Once the FairMOT baseline model is available, the next stage involves selecting a suitable VLM for pseudo-label generation in the target domain. This step involves applying the baseline in the target domain using a low confidence threshold of just 0.01. As previously mentioned, this is done because many true airplane instances receive low confidence scores under domain shift, which is why lowering the threshold is necessary in order to retain all possible candidates in the target domain. Bounding-box crops are then extracted from these candidate detections and normalized. A manually labeled airplane classification dataset is then constructed from these crops. This dataset is subsequently used to evaluate three different VLMs in terms of their ability to distinguish airplane from non-airplane crops. The selection criterion for this VLM benchmark was therefore chosen to be the classification accuracy. Once the VLM with the highest accuracy was identified, all necessary preparations for the final stage of this work were complete. This VLM benchmarking pipeline is further visualized in figure 2.

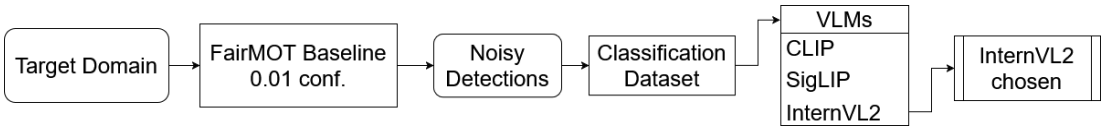


Figure 2: Overview of the proposed VLM benchmarking pipeline

Once the pre-trained baseline model had been created and the suitable VLM had been selected, the adaptation of the FairMOT baseline to the target domain using VLM-guided self-training could begin. This pipeline can be split into an inference phase and a training phase, as illustrated in Figure 3. The inference phase begins by applying the FairMOT baseline to the target domain with a confidence threshold of 0.01 in order to generate target domain candidate detections and trajectories. In the next step, sampled trajectory crops are evaluated

by the selected VLM, which results in VLM-validated target domain pseudo-labels. These pseudo-labels are then used to retrain the FairMOT baseline model, resulting in the adapted FairMOT model.

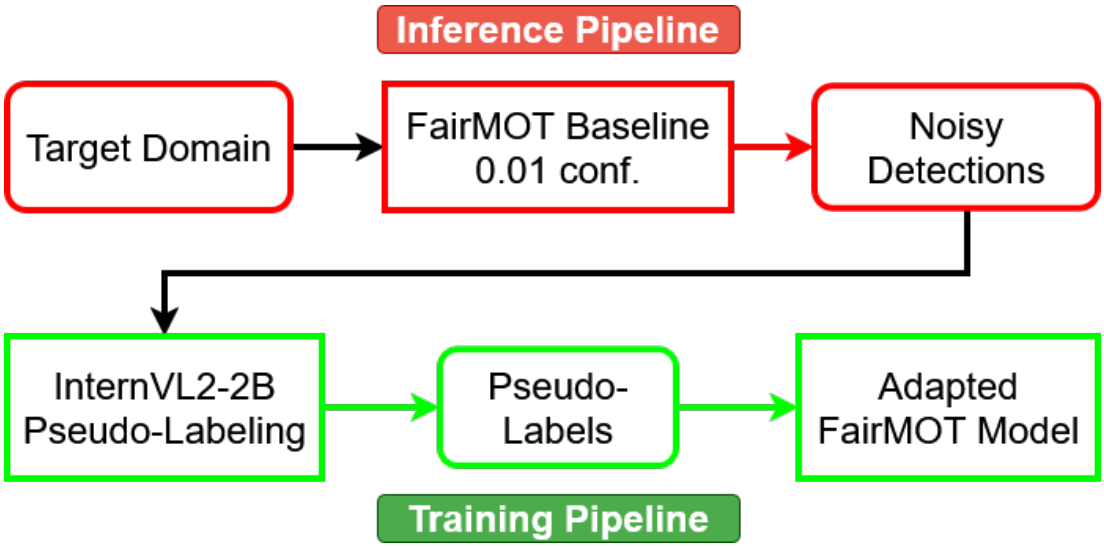


Figure 3: Overview of the proposed adaptation pipeline

3.2 MOT Datasets

This section goes into detail on dataset analysis because of the importance of properly labeled data for training purposes. A special focus is set on the chosen dataset used throughout this thesis. It explains the reasons why this data is suitable for the purpose of detecting and tracking airplanes, describes the metadata of its sequences and outlines the various adjustments that were required to correct and standardize its labels. In addition, this section also describes the chosen dataset split which was later used to train the baseline FairMOT model.

3.2.1 Dataset Overview

To prepare for model training, extensive analysis of potential datasets was conducted to find a selection of suitable datasets which could then be used for training the MOT baseline. In order for data to be fit for MOT training, multiple criteria had to be met, including but not limited to:

- surveillance-like point-of-view
- multiple aircraft instances per frame
- varying scene conditions
- MOTChallenge-format labels [9]

These criteria were enough to rule out many of the following datasets which were found and considered for use, the first of which was the TartanAviation [16] dataset. This dataset initially appeared suitable for the thesis, as it is open-source and multimodal, providing labeled RGB video frames, aircraft trajectory data, weather metadata and audio recordings of air traffic control (ATC) radio transmissions. Although the multimodal nature of this dataset could provide signals for varying approaches to domain adaptation, the main focus of this thesis was on visual data due to the planned application of VLMs, therefore only the image data of the TartanAviation set was considered. Upon closer inspection, the dataset was lacking in two main areas: first of all, the visual data contained only two differing viewpoints, which would be insufficient for testing

Sequence	Frames	FPS	Resolution	Potential Challenges
1	3 605	25	1280×960	partial occlusion, heat haze
2	3 605	25	1280×960	partial occlusion, clear weather
3	4 625	25	1280×960	partial occlusion, cloudy
4	7 299	25	1280×960	large foreground, clear weather
5	4 311	25	1280×960	partial occlusion, distant airplanes, cloudy
6	3 377	25	1280×960	partial occlusion, cloudy, many vehicles
7	2 484	25	1280×720	heat haze, multiple moving airplanes
8	2 751	25	1280×720	heat haze, distant airplanes
9	3 259	25	1920×1080	heat haze, multiple moving airplanes
10	4 197	25	1920×1080	heat haze, close aircraft interactions
Total	39 513			

Table 1: AGVS-T24 sequence metadata

domain adaptation. Secondly, most frames contain only a single airborne aircraft, which is ill-suited for MOT training. Moreover, the intended deployment setting of the tracker is airport surveillance footage, where aircraft are most commonly observed while taxiing rather than in flight. For these reasons, the TartanAviation dataset was judged as unfit for the goals of this thesis.

Another dataset which was briefly considered but quickly disregarded is the HRPlanesV2 [24] dataset, as it was used in a number of different scientific works [12] [18] for airplane detection in imagery. The reason for the quick disqualification of this dataset was the viewpoint, as the images consist of a top-down satellite view of airports, which differs substantially from the needed surveillance-like point-of-view.

In addition to TartanAviation and HRPlanesV2, multiple datasets of the AGVS-CAAC airport ground surveillance dataset collection [5] were considered. One such dataset is AVSS [7]. This set consisted of 250 videos totaling 140 000 frames, which would be a suitable amount of data for training. Upon closer inspection it was soon realized that the ground truth annotations of the data were created for semantic segmentation of the frames instead of the MOTChallenge format which was required for this thesis.

Another dataset of the AGVS-CAAC collection considered was the AFTD [8] dataset. The creators of the dataset explicitly mention that the labels were created manually and represent "precise bounding boxes for movable targets on the airport ground" [8]. They specifically stress the fact that this dataset contains multi-scale, multi-level occlusion and viewpoint changes, which would be well suited for training a robust airport surveillance MOT model. The dataset was ultimately not used due to limited availability at the time of the study and because it overlaps substantially with a more suitable dataset selected for the final experiments. To avoid redundancy and ensure a consistent training source, the subsequent dataset was preferred.

3.2.2 The AGVS-T24 Dataset

The AGVS-T24 dataset [6] is one of the two official subsets of the AGVS-T benchmark, the object tracking branch of the AGVS [30] dataset family, which stands for "Airport Ground Video Surveillance". This dataset was introduced by Zhang et al. for the express purpose of benchmarking multi-object detection and tracking models for use in airplane tracking scenarios. According to the official AGVS-T24 description [6], the dataset contains 51 long video sequences, 150 000 frames and covers a wide range of aircraft motion patterns, including takeoff, landing, and taxiing. It also includes challenges such as visual background variation, multi-scale targets and differing lighting and weather conditions, making it perfectly suitable for testing domain adaptation.

From the 51 sequences of AGVS-T24, a subset was selected as the dataset for this thesis. The selected subset consists of 10 sequences covering a diverse range of airplane motion patterns and environmental conditions. To obtain a better overview of the dataset, the sequences were thoroughly inspected and their metadata was recorded after the required repairs to the ground-truth annotations had been applied. In addition, the dataset does not provide field of view information or frame rate metadata. Therefore, all sequences were processed at 25 FPS for the experiments in this thesis. This frame rate was chosen as a reasonable approximation based on the apparent motion of taxiing aircraft. Table 1 summarizes the metadata of the selected sequences, while Figure 4 shows the first frame of each sequence used in this thesis.

Figure 4: Representative frames from the 10 AGVS-T24 sequences. Ordered left-to-right, then top-to-bottom: Sequence 1 to 10

3.2.3 Dataset Issues and Pre-Processing

Before the AGVS-T24 data could be used for training, it needed to be analyzed to ensure that the directory layout and annotation format followed the standard conventions of the MOTChallenge benchmark. Each sequence should consist of an `img1/` directory containing all frames named using a six-digit JPEG naming convention, where the starting frame is named `000001.jpg`. Additionally alongside the `img1/` directory lies the `gt/` directory containing a single `gt.txt` file. This file contains all labels of the sequence, with each individual line describing a single bounding box of a single frame, consisting of nine comma-separated columns: frame number, identity number, bounding box left, bounding box top, bounding box width, bounding box height, confidence score, class, and visibility. A single `seqinfo.ini` file is be placed inside each sequence root directory describing the sequences' metadata including resolution and frame rate.

To evaluate label correctness and identify inconsistencies, the annotations were first drawn onto the corresponding frames and the resulting visualizations were then compiled into `.mp4` videos using `ffmpeg`, which allowed for efficient manual inspection. This process revealed multiple issues such as missing annotations, corrupted frames, misaligned labels and inconsistent identity assignments. These problems needed to be addressed before training of the baseline FairMOT model could commence. The following subsections describe each issue in detail alongside the necessary pre-processing steps required to repair them.

Missing Labels in the Final Frames

During the visual inspection, it became apparent that some sequences contained segments with missing bounding-box annotations. This absence of proper ground-truth information would introduce incorrect signals during training. To prevent this, the affected frames were removed entirely from the sequence, as this solution was both straightforward and resulted in minimal data loss. Figure 5 illustrates a frame where the missing annotations result in a plain image with no bounding boxes.

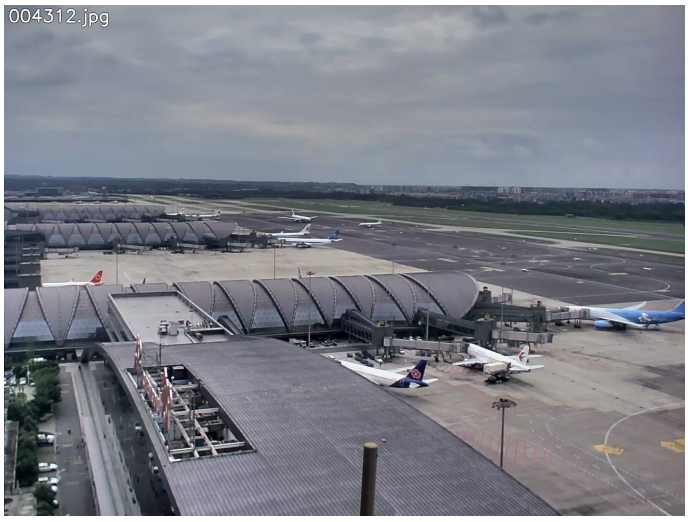
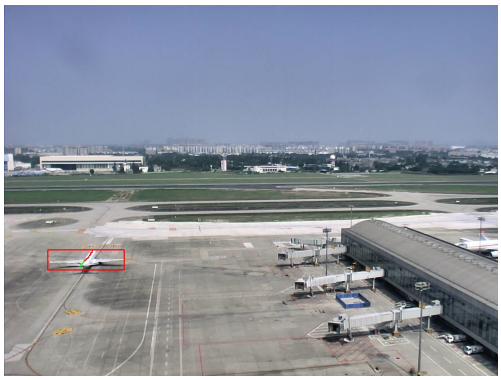


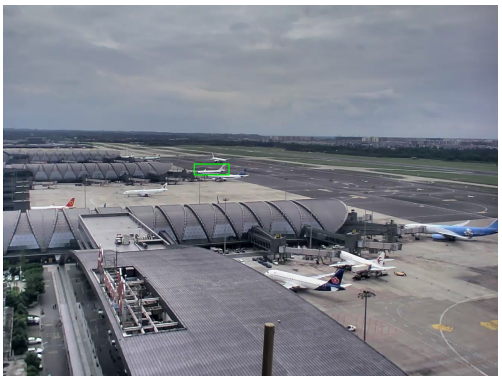
Figure 5: Frame 4312 of Sequence 5, missing all annotations

Corrupted and Processed Frames

Upon further visual inspection of the video files, it was noticed that some sequences contained processed frames. These frames contained drawn airplane trajectories which would occasionally appear during video playback. To resolve this issue, each `img1/` directory was checked and any corrupted images or files were removed. Figure 6 displays an example of a processed frame from each affected sequence.



(a) Sequence 4



(b) Sequence 5

(c) Sequence 8

Figure 6: Examples of processed frames found in Sequences 4, 5, and 8, containing drawn airplane trajectories.

Label-frame Misalignment

It was also discovered that for one sequence, the video would suddenly revert to an earlier part of the sequence during playback. This caused the bounding boxes to become misaligned with the visual placement of the airplanes. This sudden inconsistency of object positions and temporal continuity would negatively affect model training. For this reason, all frames and annotations starting from the beginning of the affected segment were removed from the sequence. Figure 7 shows a displaced frame with misaligned bounding boxes.

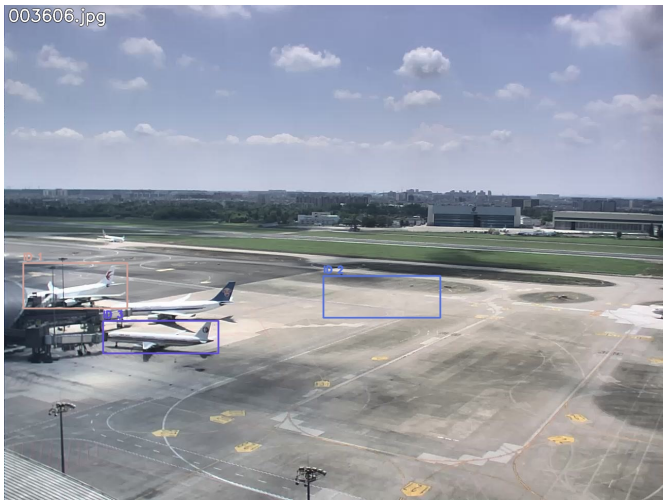


Figure 7: A frame from Sequence 2 containing a misaligned bounding box.

Non-persistent Tracking IDs

The most severe issue was the non-persistence of tracking IDs. All sequences were affected by this issue most likely due to the original intentions behind the manual annotations. Instead of each airplane instance keeping its unique ID through its entire lifetime, the ground-truth data would instead use the IDs to count the number of airplanes present inside a frame at any given moment. This would lead to airplanes incrementing or decrementing their IDs depending on whether an airplane entered or left the frame. This approach to ID assignment is ill-suited for object tracking, because it destroys the temporal consistency required for MOT training.

This problem was solved by writing Python code which enabled semi-automatic swapping of IDs in a controllable manner. Using this code together with manual inspection of the videos, it was possible to parse the `gt.txt` files and switch the IDs back in order to correctly track each unique airplane instance.

(a) Sequence 5 frame 410

(b) Sequence 5 frame 510

Figure 8: Examples of ID switches occurring in the original Sequence 5

3.2.4 Dataset Split

Before training the FairMOT baseline, the AGVS-T24 dataset was partitioned into a source domain for training and a withheld target domain for evaluating domain shift. Sequences 1, 2 and 3 were designated as the target domain because they share a common viewpoint and therefore form a consistent, unseen setting. In order to prevent information leakage, a sequence-level split was adopted, consistent with common practice in domain adaptation studies [10] and similar to the approach used for UA-DETRAC [26]. This means that each sequence was assigned entirely to either training, validation or testing. It should be noted that the validation and test sequences were chosen not only to achieve the desired split ratio, but also because they exhibit noticeable visual differences relative to the training sequences. As a consequence, performance may be lower on the validation and test sequences, providing a conservative estimate of the model's real-world performance and generalization capabilities. The resulting dataset split is summarized in Table 2.

Domain	**Split**	**Sequences**	**Frames**	**Share**
Target	Eval	1, 2, 3	11 835	100 %
Source	Train	4, 5, 6, 9, 10	22 443	81 %
Source	Val	7	2 484	9 %
Source	Test	8	2 751	10 %

Table 2: AGVS-T24 dataset split

(a) Sequence 1

(b) Sequence 2

(c) Sequence 3

Figure 9: Example frames of the new, unseen domain serving as the target for the domain adaptation experiment

3.3 FairMOT Baseline

A FairMOT baseline model was used as the starting point for the domain adaptation experiment and as the reference model for later comparison with the adapted model. Rather than training the network from scratch, the FairMOT model with a DLA-34 backbone [28] (`fairmot_dla34.pth`) was used as the initial checkpoint. DLA-34 is the default backbone of the FairMOT framework and the provided checkpoint is pre-trained on CrowdHuman [21]. Starting from this checkpoint, the FairMOT network was fine-tuned on the AGVS-T24 source domain split described previously in order to adapt the pedestrian-oriented baseline to airplane detection and tracking.

FairMOT was chosen for this experiment because it is an online one-shot multi-object tracker in which object detection and identity embedding are produced jointly from a shared backbone. This makes it a suitable baseline for studying domain adaptation, since the same tracking architecture can first be evaluated on the unseen target domain and can then be adapted through retraining on VLM-validated pseudo-labels without modifying the deployed FairMOT architecture itself. In this way, the baseline model serves both as the initial tracker used for target domain candidate trajectory generation and as the model that is later retrained to obtain the adapted FairMOT variant. Figure 10 provides an overview of the architecture.

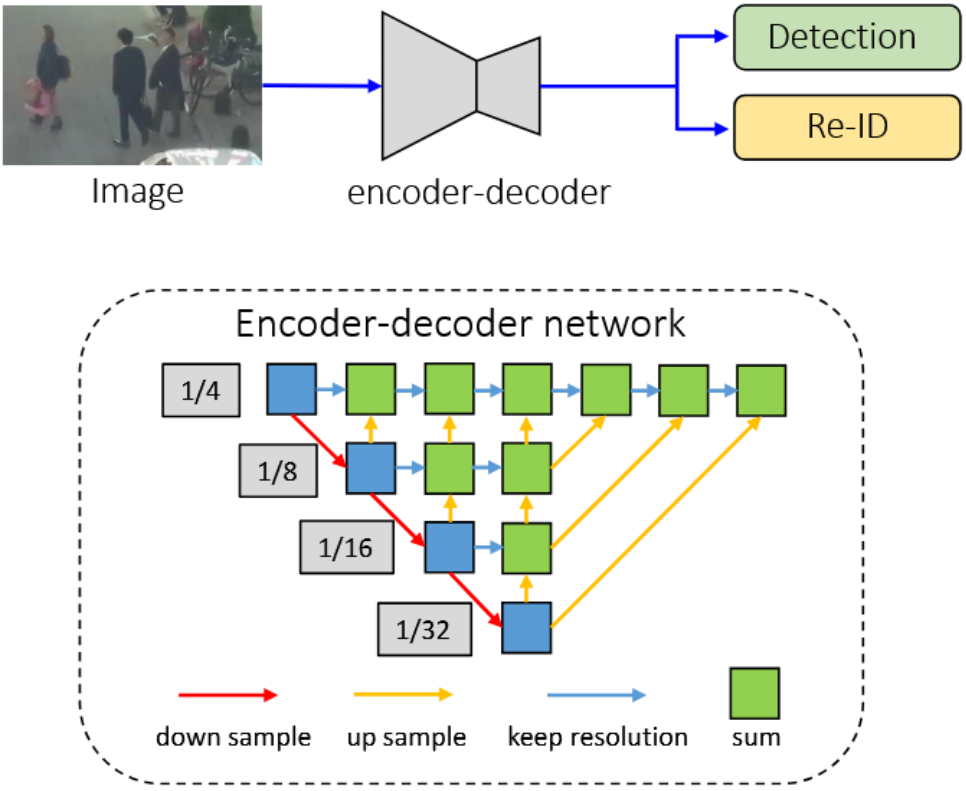


Figure 10: FairMOT architecture overview (reproduced from Zhang et al. [31])

3.4 VLM Selection and Optimization

Before the selected vision-language model can be integrated into the adaptation pipeline, its suitability for target domain pseudo-label generation must first be evaluated. To this end, a custom classification dataset was constructed from bounding-box crops extracted from target domain candidate detections produce by the FairMOT baseline. Candidate VLMs were then compared on this dataset in order to determine which model is most suitable for distinguishing airplane from non-airplane crops under domain shift. In addition to model selection, prompt

design was also considered, since different VLM families require different input formulations and may respond differently to variations in label phrasing.

3.4.1 VLM Benchmarking

To evaluate the performance of the considered vision-language models, a custom classification dataset was constructed from target-domain detections. For this purpose, Sequences 1, 2, and 3 were processed with the baseline FairMOT model using a low confidence threshold in order to retain a broad set of candidate detections, including both airplane instances and false positives. The resulting bounding-box crops were manually labeled and balanced across the airplane and non-airplane classes. The final dataset consisted of 1000 normalized image crops, with 500 airplane and 500 non-airplane samples. Because the dataset was generated from target domain candidate detections produced by the baseline model, it reflects the crop characteristics that the VLM must later handle during pseudo-label generation.

Aspect	**Description**
Data source	Sequences 1, 2, 3
Crop generation	FairMOT baseline detections at low confidence threshold
Annotation	Manual binary labels (airplane / non-airplane)
Dataset size	1000 images
Class balance	500 airplanes, 500 non-airplanes

Table 3: Summary of the classification dataset used for VLM benchmarking

After the dataset had been created, VLM model selection was performed. The aim was to evaluate CLIP, SigLIP, and InternVL2 on the same benchmark in order to determine which model was most suitable for target-domain pseudo-label generation in the FairMOT adaptation pipeline. Each VLM was applied to the same airplane dataset, and its output was mapped to a binary airplane versus non-airplane decision. For the contrastive models, this decision was derived from similarity-based scores, whereas InternVL2 returned a text label directly through natural-language generation.

After evaluating all three VLMs on the custom dataset, InternVL2 achieved the strongest classification performance and was therefore selected for the subsequent adaptation pipeline. The quantitative comparison underlying this choice is presented in Section 4.3.

3.4.2 Prompt Engineering and Variation

Prompt engineering is an important aspect of VLM evaluation, which is why care was taken to formulate prompts that were appropriate for the respective model families. The expected input formats of CLIP and SigLIP differ from those of InternVL2, which required different prompting strategies. For the contrastive models, the classification task was formulated by using a fixed label set and a shared prompt template. In this setting, the airplane-related label was treated as the positive class, while all remaining labels were treated as non-airplanes.

For InternVL2, the classification task was instead formulated as a natural-language instruction. Rather than using the same prompt structure as the contrastive models, the candidate labels were embedded directly into a textual prompt and the model was instructed to return exactly one category. During preliminary experimentation, it was observed that small changes in label phrasing could noticeably affect classification behavior. This motivated a more careful prompt design for InternVL2 and highlighted the general sensitivity of VLMs to prompt wording.

After InternVL2 was selected for the subsequent adaptation pipeline, two prompting strategies were further considered in order to assess whether prompt wording also affects behavior

under tracking conditions. The first strategy retained the multi-class formulation used during the classification benchmark, whereas the second reduced the decision to a binary yes/no question. In both cases, the VLM output was interpreted as a binary airplane versus non-airplane decision. The comparative effect of these prompt variants is analyzed later in Section 4.3.

3.5 FairMOT Adaptation

After a suitable VLM has been selected, it is incorporated into the final stage of the proposed method in order to adapt the FairMOT baseline to the unseen target domain. This adaptation is performed by generating target domain pseudo-labels with the help of the selected VLM and then retraining the baseline FairMOT model on these pseudo-labels. In this way, the model is exposed to the target domain during training without requiring manual annotation. The following subsection describes the VLM-guided pseudo-label generation procedure and the subsequent retraining step used to produce the adapted FairMOT model.

3.5.1 VLM-Guided Pseudo-Label Generation

After the classification benchmark and VLM selection were completed, the chosen VLM was integrated into the target domain adaptation pipeline in order to generate pseudo-labels for subsequent retraining. To obtain a reference point, the baseline FairMOT model was first evaluated on the unseen target domain using the standard tracking configuration. Because the original FairMOT framework was designed for pedestrians, its pipeline had to be adjusted to account for airplane-specific bounding-box characteristics. In particular, the tracking configuration was adapted so that plausible airplane detections were retained instead of being suppressed by assumptions tailored to pedestrian-like object shapes.

For pseudo-label generation, the baseline model was then applied to the target domain with a substantially lower confidence threshold in order to retain a broad set of target domain candidate detections. These detections were grouped by track ID and treated as trajectories rather than as isolated frame-wise observations.

Formally, a candidate trajectory τ_k is defined as the ordered set of detections assigned to track ID k:

$$\tau_k = \{(f_i, \mathbf{b}_i)\}_{i=1}^{L_k}, \tag{1}$$

where f_i denotes the frame index, $\mathbf{b}_i = (x_i, y_i, w_i, h_i)$ denotes the corresponding bounding box in MOT format, and L_k is the length of the trajectory.

For VLM validation, only detections whose temporal position lies within the central 50% of the trajectory were considered, while the first and last quarters of the trajectory were discarded because they more frequently contain partial views or unstable bounding boxes.

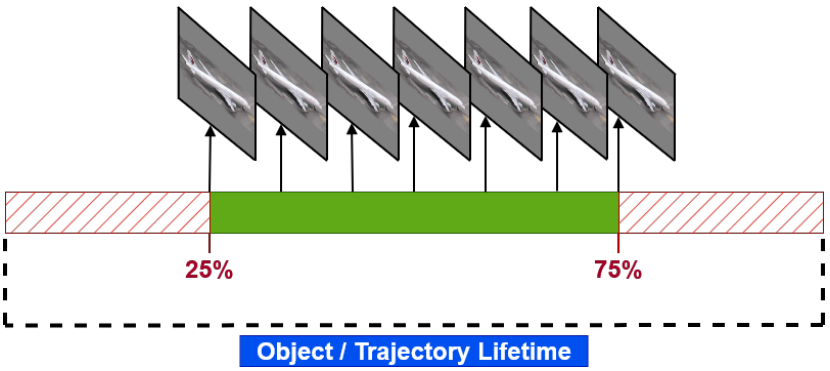


Figure 11: Proposed majority voting strategy

Within this region of interest, seven uniformly spaced bounding-box crops were extracted and presented to the VLM. The resulting predictions were then aggregated using majority voting. In this way, each candidate trajectory was classified as either airplane or non-airplane based on repeated observations over time rather than a single frame. Trajectories validated as airplanes were then converted into target domain pseudo-labels in MOTChallenge format. This trajectory-based validation strategy is computationally more efficient than a naive frame-by-frame VLM evaluation. In the naive case, the VLM would have to process every detected bounding box in every frame of a trajectory. For a trajectory of length L_k, this corresponds to L_k VLM evaluations. In contrast, the proposed method evaluates only seven sampled crops per trajectory, independent of its full length. Despite this reduction in VLM evaluations, the trajectory-level aggregation still provides a robust validation signal.

These VLM-validated pseudo-labels were subsequently used to retrain the baseline FairMOT model on target domain data, resulting in the adapted FairMOT model. The quantitative impact of this pseudo-label generation and retraining procedure on tracking performance is discussed later in Section 4.6.

4 Experiment Results

This section presents the results of all conducted experiments. It covers the evaluation protocol and metrics used, the results of the VLM classification experiment, summarizes the target domain tracking results obtained from VLM-guided pseudo-label generation and retraining, and presents a quantitative analysis of the findings. For readability, the main findings are briefly summarized in text and the full numerical data is provided in tables.

4.1 Tracking Evaluator and Metrics

In order to evaluate and analyze the target domain tracking results, FairMOT's evaluation module (`src/lib/tracking_utils/evaluation.py`) was used. This evaluator generates MOTChallenge-style metrics using the `py-motmetrics` library, where predicted and ground-truth bounding boxes are matched using IoU-based assignment. A match is only considered valid if the intersection-over-union satisfies the threshold 0.5, otherwise the prediction counts as a false positive and the ground-truth object as a miss. The tracker results were exported in MOTChallenge format and compared against the ground-truth labels on a frame-by-frame basis. Throughout Section 4, the metrics IDF1, Precision, Recall, MOTA, MOTP, FP, FN and IDS are reported, as these metrics provide valuable insight into the performance of the tracker. These metrics can be briefly explained as:

- **Precision:** fraction of reported detections that are correct matches
- **Recall:** fraction of ground-truth objects that are recovered by the tracker
- **FP (false positives):** number of predicted detections that do not match the ground-truth
- **FN (false negatives):** number of ground-truth objects not matched by any prediction
- **IDS (identity switches):** count of times the ground-truth changes the matched ID across frames
- **MOTA:** overall tracking accuracy that penalizes FP, FN, and IDS relative to the number of ground-truth objects
- **MOTP:** localization precision over matched pairs, computed as the average localization error
- **IDF1:** identity F1 score measuring ID consistency over time

4.2 Baseline Training

The FairMOT DLA-34 model was fine-tuned on the AGVS-T24 dataset split in order to obtain the baseline model. The resulting checkpoint is referred to as `fairmot_dla34_AGVST24_e30.pth`. Training was performed on a single GPU with 8 GB of VRAM, which constrained the batch size accordingly. Training progress was monitored with Weights & Biases (W&B) [2] to track the individual loss terms and check for instability during training. In order to keep training diagnostics easy to interpret, fixed multi-task loss weights were used instead of the default uncertainty-based weighting. The default weights could yield negative aggregated loss values, which complicates reading and comparing loss curves. Low-level execution details such as the exact command-line arguments, the configuration file and the environment setup are provided in the appendix.

4.2.1 Loss Curves and Training Stability

Training stability was assessed using the loss curves recorded over the 30-epoch run. The FairMOT objective combines losses for object center prediction, bounding-box regression and an identity embedding loss for re-identification along with an aggregated total loss. Across the entire training run, each loss term decreased consistently and the total loss converged towards the later epochs, as can be seen in Figure 12. This indicates stable optimization without divergence. The full set of loss curves is shown in the appendix. The resulting `fairmot_dla34_AGVST24_e30.pth` checkpoint serves as the FairMOT baseline for the subsequent domain adaptation experiment.

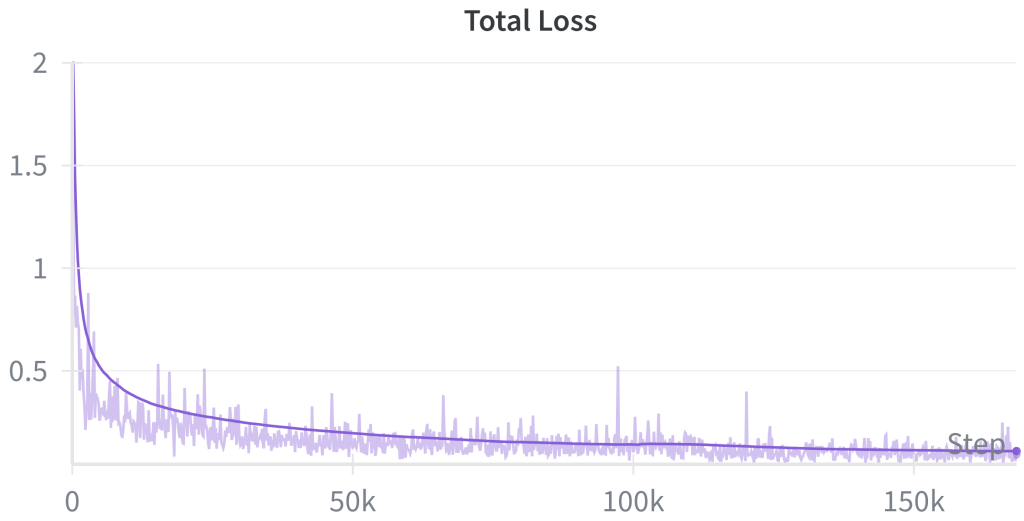


Figure 12: Total Loss

4.2.2 Baseline Validation and Testing

Although validation was scheduled after each epoch via the `val_intervals` flag in the training command, no reliable validation metrics were produced in the training logs for the reported run. As a consequence of this, no validation-based early stopping or checkpoint selection could be performed. Instead, the final checkpoint after 30 epochs was selected as the baseline model. This checkpoint was subsequently evaluated on the held-out validation and test sequences by running the tracker of the FairMOT framework (`src/track.py`) to generate the metrics and

comparison to the ground truth. The model was run on these sequences using the default confidence threshold of 0.4, as is recommended by the authors of FairMOT. The results of this manual validation and testing showed that the trained model did lose some accuracy on the unseen validation and test sequences, however this was to be expected. The results still proved good enough to further use the model in the continuation of this thesis. The validation and test results are provided in Table 4 and 5 respectively.

Sequence	IDF1	Precision	Recall	MOTA	MOTP	FP	FN	IDS
seq-07	0.646	0.973	0.665	0.645	0.229	202	3 615	10

Table 4: FairMOT baseline validation results

Sequence	IDF1	Precision	Recall	MOTA	MOTP	FP	FN	IDS
seq-08	0.325	0.654	0.341	0.159	0.323	5 618	20 502	40

Table 5: FairMOT baseline testing results

4.3 Classification Results

As a preliminary step before the target-domain adaptation experiment, a classification benchmark was performed in order to select the most suitable vision-language model for airplane classification. The three candidate VLMs were evaluated on the custom airplane dataset described in Section 3.4.1. For each model, the number of correct and incorrect classifications as well as the overall classification accuracy were recorded. Based on the highest classification accuracy, InternVL2-2B was selected for the subsequent pseudo-label generation stage.

Although all three VLMs were evaluated on the same dataset, their input formulations differed due to architectural differences between contrastive and generative vision-language models. CLIP and SigLIP were evaluated using a shared prompt template together with a fixed multi-class label set. The template used for both models was:

```
"It is a photo of {label}."
```

The label set consisted of the following categories:

```
dominant airplane,
airport environment,
service vehicle,
passenger bus,
paving,
building,
open field,
sky,
parking lot,
background,
distant object
```

For the final binary evaluation, the label `dominant airplane` was interpreted as the positive airplane class, while all remaining labels were mapped to the non-airplane class.

InternVL2 was evaluated differently, since it operates through natural-language prompting rather than a contrastive label-matching interface. Instead of the fixed template used for CLIP and SigLIP, InternVL2 was given a natural-language instruction containing the candidate categories explicitly. The prompt used for InternVL2 was:

```
"You are an image classifier for airport surveillance images.
Choose the single best category for the image from the following list:
aircraft, airport environment, service vehicle, passenger bus, paving,
building, open field, sky, parking lot, background, distant object.
Answer with exactly one category name from the list, with no extra words."
```

A small but relevant prompt adjustment was made for InternVL2: the category `dominant airplane` used for CLIP and SigLIP was replaced with the shorter label `aircraft`, since preliminary testing showed that this wording led to better classification behavior for InternVL2. Unlike CLIP and SigLIP, InternVL2 does not return logit vectors that can be interpreted through sigmoid or softmax activations. Its outputs were therefore treated purely as discrete class labels extracted from the generated natural-language response.

In addition to prompt formulation, the exact model instances also differed. CLIP was evaluated using openai/clip-vit-large-patch14-336, SigLIP using google/siglip-so400m-patch14-224, and InternVL2 using OpenGVLab/InternVL2-2B. The 2B InternVL2 variant was chosen because it was compatible with the available 8 GB VRAM budget while still providing strong multimodal classification performance.

The overall quantitative comparison is shown in Table 6. InternVL2-2B achieved the highest classification accuracy with 937 correct predictions out of 1000 samples, corresponding to 93.7%. SigLIP followed with 91.5%, while CLIP reached 88.3%. Based on this result, InternVL2-2B was selected for the subsequent target-domain pseudo-label generation pipeline.

Model	Instance	Correct	Incorrect	Accuracy
InternVL2	`OpenGVLab/InternVL2-2B`	937	63	93.7%
SigLIP	`google/siglip-so400m-patch14-224`	915	85	91.5%
CLIP	`openai/clip-vit-large-patch14-336`	883	117	88.3%

Table 6: Overall VLM airplane classification accuracy

Figures 13, 14 and 15 further illustrate representative classification outputs for the three evaluated VLMs. These qualitative examples provide additional insight into the label behavior of each model and complement the quantitative comparison shown in Table 6.

Figure 13: Representative InternVL2 detection samples for the predicted labels. Green frames indicate correctly classified crop samples, while red frames indicate incorrectly classified ones.

Figure 14: Highest-scoring SigLIP detection samples representative of each label. Green frames indicate correctly classified crop samples, while red frames indicate incorrectly classified ones.

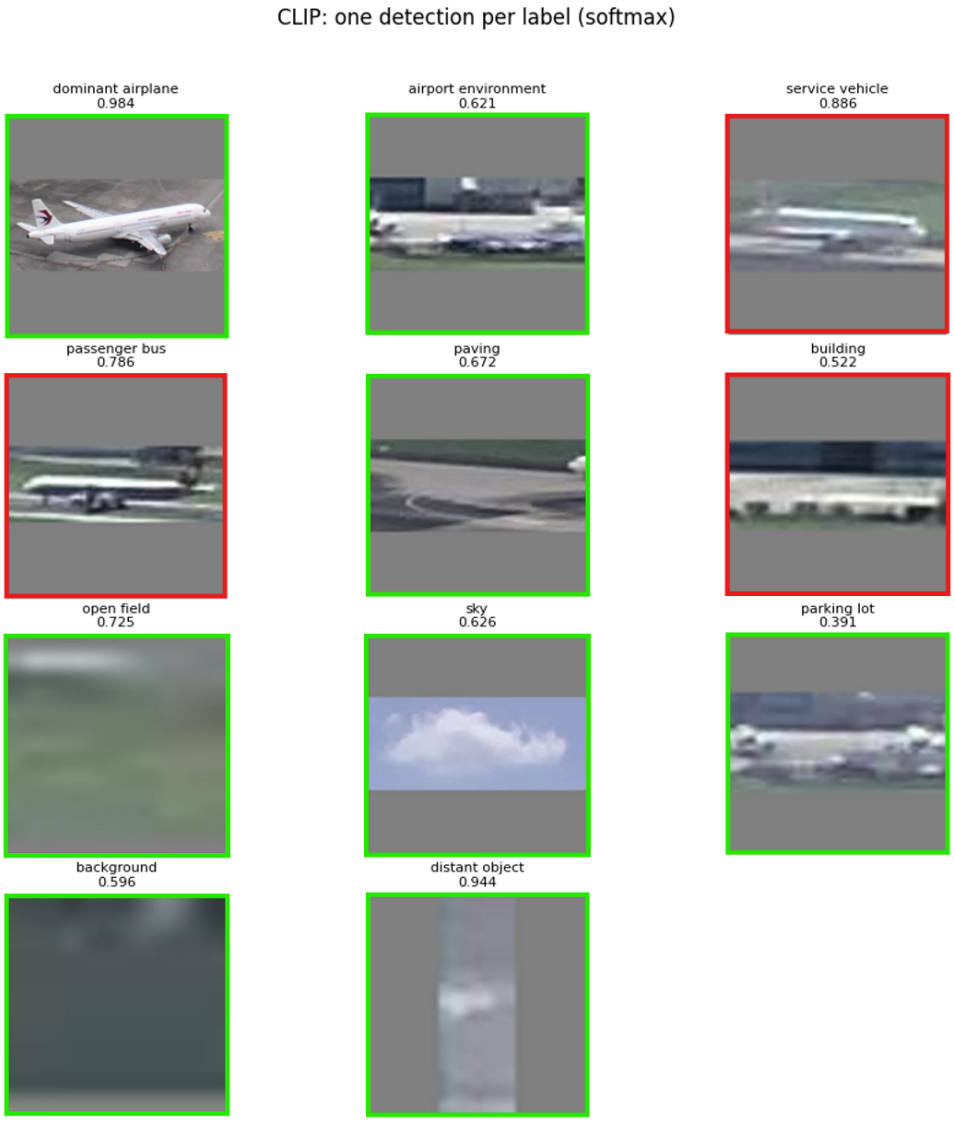


Figure 15: Highest-scoring CLIP detection samples representative of each label. Green frames indicate correctly classified crop samples, while red frames indicate incorrectly classified ones.

After InternVL2-2B had been selected through the classification benchmark, two prompt variants were considered for its later use in the target-domain pseudo-label generation pipeline. The first retained the multi-class formulation used during benchmarking, while the second reduced the task to a binary yes/no question. The binary prompt used was:

```
"Is this an image of an aircraft? Answer yes or no."
```

The motivation behind this comparison was to assess whether prompt wording and output formulation also affect performance in the subsequent tracking setting. The quantitative comparison of these two InternVL2 prompt variants is presented later in Section 4.5.

4.4 FairMOT Adaptation via Retraining

The retraining step uses the VLM-validated target domain pseudo-labels generated by the InternVL2-based run described in Section 3.5.1 as training supervision. In particular, the pseudo-labels of the multi-class prompt variant were used, as they provided better overall

metrics compared to the binary prompt. Further details about this performance difference are described in the Section 4.5. The VLM-validated pseudo-labels in MOTChallenge format were converted into a pseudo-label training set in the format expected by FairMOT, enabling standard fine-tuning without the need for changing the model architecture or training code. Starting from the baseline checkpoint `fairmot_dla34_AGVST24_e30.pth`, the model was fine-tuned for 10 epochs on the pseudo-labeled target domain data using the same backbone architecture and loss configuration as in the baseline training. As before, training was monitored using Weights & Biases and fixed multi-task loss weights were used to keep loss curves interpretable. Low-level execution details such as configuration files and command-line arguments are provided in the appendix.

4.4.1 Loss Curves and Training Stability

Training stability of the pseudo-label retraining run was assessed using the loss curves recorded over the 10-epoch fine-tuning run. The aggregated training loss decreased consistently throughout the run and the individual loss components remained stable, indicating optimization without divergence. Figure 16 shows the total loss during the retraining of the FairMOT baseline. The full set of refinement loss curves is provided in the appendix. The effect of retraining on VLM-validated pseudo-labels on target domain tracking performance is reported in Section 4.5.

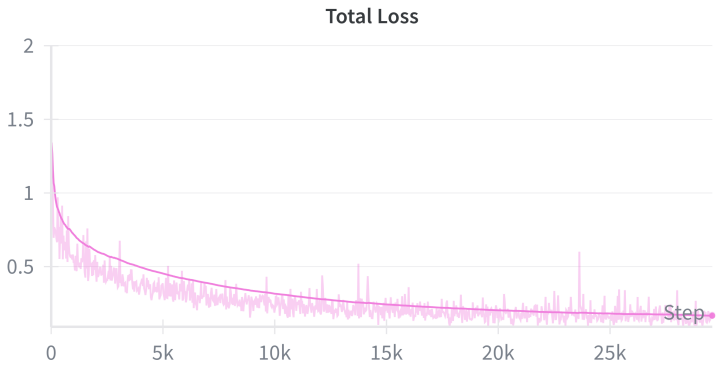


Figure 16: Total loss during retraining

4.5 Tracking Results

After selecting InternVL2-2B as the VLM for pseudo-label generation, the tracking experiment was conducted on the new domain consisting of Sequences 1 through 3. First, the pre-trained FairMOT model `fairmot_dla34_AGVST24_e30.pth` was run with the default tracking confidence threshold of 0.4 to obtain a baseline which will subsequently be used to perform a relative metric comparison. In addition, two runs were performed in which the baseline model was applied with a tracking confidence threshold of 0.01, after which InternVL2-2B was applied to generate pseudo-labels, once per prompt variant. For the sake of readability, `prompt A` refers to the multi-class prompt, and `prompt B` refers to the yes/no question, both of which are described in detail in Section 3.4.2. In addition to the direct VLM-guided pseudo-label runs, the FairMOT baseline was also fine-tuned for 10 epochs on the pseudo-labels generated by InternVL2-2B using prompt A (Section 3.4.2), and the resulting adapted checkpoint was evaluated on the same target domain sequences using the same tracking and evaluation pipeline as the baseline. The following tables report per-sequence results for each run. For clarity, the reported overall value is the arithmetic mean across the three sequences rather than the evaluator's aggregated multi-sequence `OVERALL` score. It should be noted that the first five metrics had their mean

values rounded to three decimal places, whereas the count-based metrics (FP, FN and IDS) were rounded to integers, since they represent discrete occurrences.

Sequence	IDF1	Precision	Recall	MOTA	MOTP	FP	FN	IDS
seq-01	0.643	0.968	0.498	0.481	0.234	444	13 535	20
seq-02	0.840	0.945	0.830	0.781	0.277	794	2 769	6
seq-03	0.759	0.847	0.880	0.721	0.278	4 322	3 251	7
MEAN	**0.747**	**0.920**	**0.736**	**0.661**	**0.263**	**1 853**	**6 518**	**11**

Table 7: Baseline FairMOT tracking results on the new domain

Sequence	IDF1	Precision	Recall	MOTA	MOTP	FP	FN	IDS
seq-01	0.822	0.862	0.786	0.660	0.291	3 393	5 783	0
seq-02	0.890	0.967	0.825	0.797	0.266	458	2 861	0
seq-03	0.822	0.948	0.768	0.725	0.256	1 154	6 286	13
MEAN	**0.845**	**0.926**	**0.793**	**0.727**	**0.271**	**1 668**	**4 977**	**4**

Table 8: InternVL2-2B prompt A pseudo-label metrics

Sequence	IDF1	Precision	Recall	MOTA	MOTP	FP	FN	IDS
seq-01	0.699	0.629	0.786	0.322	0.291	12 507	5 783	0
seq-02	0.827	0.765	0.900	0.623	0.274	4 519	1 626	0
seq-03	0.852	0.885	0.868	0.755	0.279	3 069	3 576	13
MEAN	**0.793**	**0.760**	**0.851**	**0.567**	**0.281**	**6 698**	**3 662**	**4**

Table 9: InternVL2-2B prompt B pseudo-label metrics

Sequence	IDF1	Precision	Recall	MOTA	MOTP	FP	FN	IDS
seq-01	0.820	0.858	0.787	0.657	0.297	3 516	5 736	4
seq-02	0.894	0.964	0.839	0.807	0.271	510	2 633	3
seq-03	0.816	0.918	0.752	0.684	0.257	1 827	6 731	3
MEAN	**0.843**	**0.913**	**0.793**	**0.716**	**0.275**	**1 951**	**5 033**	**3**

Table 10: Adapted FairMOT tracking results on the new domain

4.6 Relative Comparison and Analysis of Findings

To present the effects of VLM-guided target domain adaptation in a more readable form, this section discusses the relative metric changes of the InternVL2-based pseudo-label runs on the new domain as well as the adapted checkpoint obtained through retraining on pseudo-labels, compared to the FairMOT baseline. The values in the following table are computed from the `MEAN` row of each individual run. In the context of this relative comparison, the FairMOT baseline serves as the reference and is therefore shown as 0% for all metrics. In addition, all calculated percentages were rounded to two decimal places. The exact formula used to calculate the relative changes was:

$$\Delta\% = \frac{x_{\text{run}} - x_{\text{baseline}}}{x_{\text{baseline}}} \cdot 100 \tag{2}$$

Note that the interpretation of the sign depends on the metric being evaluated. For the metrics IDF1, Precision, Recall, MOTA and MOTP, a positive relative change denotes an improvement in performance. On the other hand, for the metrics FP, FN and IDS, a negative relative change is indicative of fewer errors and therefore increased performance. For readability, improvements are colored green and degradations are colored red.

Run	IDF1	Precision	Recall	MOTA	MOTP	FP	FN	IDS
Baseline FairMOT	0%	0%	0%	0%	0%	0%	0%	0%
InternVL2 Prompt A	+13.12%	+0.65%	+7.74%	+9.98%	+3.04%	-9.98%	-23.64%	-63.64%
InternVL2 Prompt B	+6.16%	-17.39%	+15.63%	-14.22%	+6.84%	+261.47%	-43.82%	-63.64%
Adapted FairMOT	+12.85%	-0.76%	+7.74%	+8.32%	+4.56%	+5.29%	-22.78%	-72.73%

Table 11: VLM pseudo-labels and adapted model relative to baseline performance

Table 11 shows that VLM-guided pseudo-label generation can improve target domain tracking performance relative to the non-adapted FairMOT baseline. The tracking run using InternVL2-2B with prompt A, which was the multi-class prompt, displayed the most promising results with improvements across all metrics. Most notable are the improvements in MOTA and IDF1 and the reduction in IDS, which indicate a more general increase in tracking accuracy and identity stability respectively. In addition to these metrics, the FN score decreased by nearly 24% and the recall score increased by nearly 8%. Both of these metric changes indicate more ground-truth airplanes are detected and tracked. In the case of the multi-class prompt A, InternVL2-2B was able to provide a balanced improvement in performance by yielding fewer errors and obtaining better identity consistency.

On the other hand, InternVL2-2B using prompt B degraded overall performance compared to the baseline. This becomes clear when the decrease of roughly 14% in MOTA and 17% in precision is discussed. Among all the presented metrics, the FP value for prompt B stands out as one of the only true outliers. With a roughly 261% increase in false positive detections, the performance of the underlying MOT model degraded significantly. It should be noted that prompt B was able to detect more airplanes overall, which is indicated by the nearly 16% increase in recall and roughly 44% decrease in false negative detections. Although IDS decreased substantially, it was outweighed by the explosion of the false positive detections.

In addition to direct pseudo-label evaluation runs, the adapted FairMOT model obtained by fine-tuning on the InternVL2 prompt A pseudo-labels also improved target domain tracking performance relative to the baseline. Most notable are the significant gains in IDF1 and Recall performance, as well as the sharp decrease in IDS, which was the best relative improvement out of all runs. This indicates improved identity stability and higher coverage of ground-truth airplanes. On the other hand, the adapted model showed a small decrease in average precision and a modest increase in false positives, suggesting that this model is slightly more permissive at the default confidence threshold of 0.4 compared to the baseline.

Overall it can be said that the multi-class prompt A generally outperforms the simpler yes/no question prompt B, and that pseudo-label fine-tuning transfers a substantial portion of knowledge into the tracker's weights through retraining.

5 Conclusion

In the context of computer vision, domain adaptation remains challenging for the general application of pre-trained MOT models on new domains. It is expensive or nearly impossible to obtain a large amount of high-quality labels for a new domain, which greatly limits the

application and generalization of MOT networks, as was the case for the airplane detection and tracking task tackled in this thesis. Using VLM-guided pseudo-labels, the proposed adaptation approach improved the overall detection and tracking performance. In contrast, a less suitable prompting strategy degraded the overall performance by greatly increasing false positives. The results of this experiment imply that VLM-guided pseudo-labels can be used to adapt MOT models via retraining to mitigate performance degradation caused by domain shift, but that prompting strategies strongly control the precision-recall balance. Additionally, the strong decrease in IDS suggests that the approach described in this thesis not only improves detection quality, but also improves association in the tracking pipeline. It should be noted that this thesis is not without its limitations, as only one unseen domain and one primary VLM were used for this study. In conclusion, this thesis demonstrates that VLM-guided pseudo-label generation and MOT model retraining are practical strategies for mitigating the degradation of airplane MOT performance caused by domain shift.

6 Future Work

Although the results of this thesis show VLM-guided pseudo-labeling and subsequent retraining can improve cross-domain tracking, there remains room for improving robustness of MOT models under domain shift. One logical next step would be to utilize spatial filtering via image segmentation. In this case, detections in implausible areas of the frame, such as inside buildings or non-pavement areas, could be suppressed which could in turn reduce false positives. This could be achieved either through manual polygon masks per domain, or for a more general application, semantic segmentation could be used.

Lastly, the multimodal aspect of this thesis must be considered; while this thesis uses multimodality primarily through generating pseudo-labels from natural language, datasets [16] exist which include different types of data, such as audio transmissions or weather information. These types of data could be used as contextual information to improve cross-domain tracking by establishing temporal and spatial expectations for airplanes. For example, ATC audio could be used to identify when aircraft activity is expected (temporal), and even which runways and taxiways are currently in use (spatial). These directions for future work can improve domain adaptation by either turning VLMs into a training signal, adding strong physical constraints, or strengthening identity tracking and multimodal context.

References

[1] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 3

[2] L. Biewald. Experiment tracking with weights and biases. `https://www.wandb.com`, 2020. Software, accessed March 23, 2026. 17

[3] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2016. 4

[4] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, B. Li, P. Luo, T. Lu, Y. Qiao, and J. Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24185–24198, 2024. 2, 4

[5] Civil Aviation Administration of China. Airport ground video surveillance (AGVS-CAAC). `https://www.agvs-caac.com/`. Accessed: 2025-11-25. 7

[6] Civil Aviation Administration of China. Airport ground video surveillance for object tracking24 (AGVS-T24). `https://www.agvs-caac.com/AGVS-T/agvst24.html`. Accessed: 2025-11-25. 2, 7

[7] Civil Aviation Administration of China. Airport video semantic segmentation dataset (AVSS). `https://www.agvs-caac.com/avss/avss.html`. Accessed: 2025-11-25. 7

[8] Civil Aviation Administration of China. A foreground target detection dataset for airport scenario (AFTD). `https://www.agvs-caac.com/aftd/aftd.html`. Accessed: 2025-11-25. 7

[9] P. Dendorfer, A. Ošep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé. MOTChallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision*, 129(4):845–881, 2021. 6

[10] L. Ericsson, D. Li, and T. Hospedales. Better practices for domain adaptation. In *Proceedings of the Second International Conference on Automated Machine Learning (AutoML)*, volume 224 of *Proceedings of Machine Learning Research*, pages 4/1–25, 2023. 12

[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17:1–35, 2016. 4

[12] S. El Ghazouali, A. Gucciardi, F. Venturini, N. Venturi, M. Rueegsegger, and U. Michelucci. FlightScope: An experimental comparative review of aircraft detection algorithms in satellite imagery. *Remote Sensing*, 16(24), 2024. 7

[13] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation, 2016. arXiv:1607.03516 [cs.CV]. 4

[14] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4696–4704, 2015. 3

[15] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks, 2015. arXiv:1502.02791 [cs.LG]. 3

[16] J. Patrikar, J. Dantas, B. Moon, M. Hamidi, S. Ghosh, N. Keetha, I. Higgins, A. Chandak, T. Yoneyama, and S. Scherer. Image, speech and ADS-B trajectory datasets for terminal airspace operations. *Scientific Data*, 12(1):469, 2025. 6, 25

[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 2, 4

[18] Y. Radionov, V. Kashtan, V. Hnatushenko, and O. Kazymyrenko. Aircraft detection with deep neural networks and contour-based methods. *Radio Electronics, Computer Science, Control*, pages 121–129, 2024. 7

[19] D. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979. 3

[20] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid. Joint probabilistic data association revisited. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3047–3055, 2015. 3

[21] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun. CrowdHuman: A benchmark for detecting human in a crowd. `https://arxiv.org/abs/1805.00123`, 2018. 13

[22] B. Sun and K. Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *Computer Vision – ECCV 2016 Workshops*, pages 443–450. Springer, 2016. 3

[23] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial Discriminative Domain Adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017. 4

[24] D. Unsal. HRPlanesv2 - high resolution satellite imagery for aircraft detection, 2022. Dataset. Available at: `https://doi.org/10.5281/zenodo.7331974`. Accessed: 2026-03-23. 7

[25] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2, 3

[26] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, 193:102907, 2020. 12

[27] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017. 3

[28] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2403–2412, 2018. 5, 13

[29] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023. 2, 4

[30] X. Zhang, C. Shu, S. Li, C. Wu, and Z. Liu. AGVS: A new change detection dataset for airport ground video surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):20588–20600, 2022. 7

[31] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021. 2, 3, 13

A FairMOT Baseline Execution Details

This appendix section covers implementation-level details for reproducing the FairMOT baseline described in Section 3.3. It includes the software environment, training command, the dataset configuration file and the all loss curves referenced in Section 4.2.

A.1 Environment Setup

Training was performed on the following hardware and in a conda environment with the following key versions:

1. GPU: NVIDIA GeForce RTX 3070 Ti Laptop, 8 GB VRAM
2. Python: 3.8
3. PyTorch: 1.7.1
4. TorchVision: 0.8.2
5. CUDA: 11.0
6. cuDNN: 8.0.5

A.2 Training Command

Using the environment described above, the FairMOT DLA-34 model was used as the starting checkpoint. It was trained using the following command-line arguments:

```
Python train.py mot \
    --data_cfg ../src/lib/cfg/agvs_baseline.json \
    --load_model ../models/fairmot_dla34.pth \
    --gpus 0 \
    --batch_size 4 \
    --num_workers 8 \
    --num_epochs 30 \
    --exp_id airplane_detector_and_tracker \
    --val_intervals 1 \
    --save_all \
    --arch dla_34 \
    --multi_loss fix \
    --lr 1e-4
```

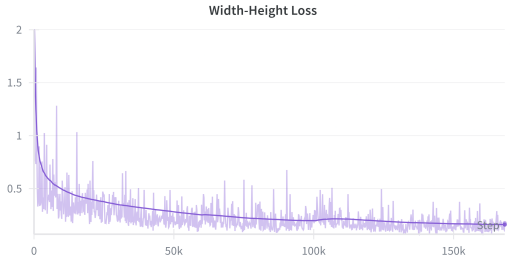
A.3 Data Configuration File

The `agvs_baseline.json` file used in the training command defines the absolute path to the root directory of the dataset split, alongside relative paths to the training, validation and test sets. The configuration file looks as follows:

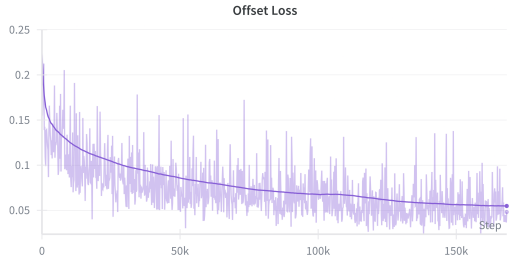
```
{
    "root": "/path/to/data/Training-AGVS-T24",
    "train": {
        "agvs": "./data/agvs.train"
    },
    "val": {
        "agvs": "./data/agvs.val"
    },
    "test": {
        "agvs": "./data/agvs.test"
    }
}
```

A.4 Loss Curves

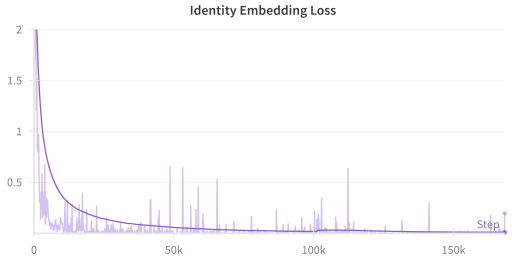
The subsequent graphs show the loss curves of the individual loss terms as they evolved over the 30-epoch training run of the FairMOT baseline. The width-height loss measures the error in the predicted bounding-box size. The offset loss reflects how accurately the model predicts the bounding-box center position compared to the ground truth. The heatmap loss measures how well the model detects the presence and location of object centers. The identity embedding loss reflects how consistently the model assigns the same identity-related features to the same target across frames.



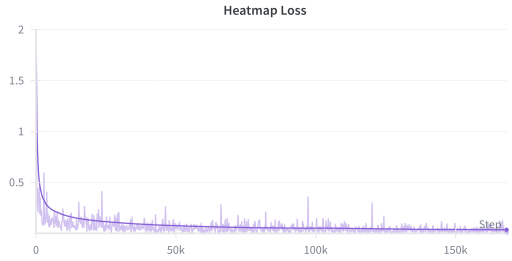
(a) Width-height loss



(b) Offset loss



(c) Identity embedding loss



(d) Heatmap loss

B FairMOT Adaptation via Retraining Details

This appendix section covers implementation-level details for reproducing the pseudo-label self-training described in Section 4.4. It includes the software environment, training command, the dataset configuration file and all loss curves.

B.1 Environment Setup

The retraining run was performed using the same hardware and software environment as the baseline training, unless noted otherwise.

B.2 Training Command

Starting from the baseline checkpoint `fairmot_dla34_AGVST24_e30.pth`, the model was fine-tuned on the pseudo-labeled target domain data using the following command-line arguments:

```
python train.py mot \
    --data_cfg ../src/lib/cfg/pseudo_labels.json \
    --load_model ../models/fairmot_dla34_AGVST24_e30.pth \
    --gpus 0 \
    --batch_size 4 \
    --num_workers 8 \
    --num_epochs 10 \
    --exp_id airplane_detector_and_tracker_pseudo \
```

```
    --arch dla_34 \
    --multi_loss fix \
    --lr 1e-5
```

B.3 Data Configuration File

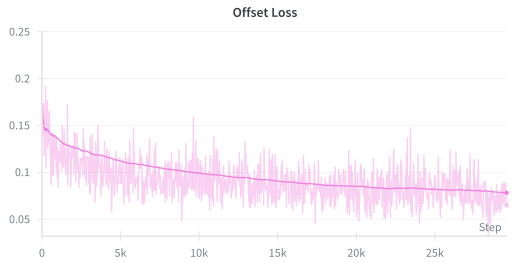
The `pseudo_labels.json` file used in the refinement command defines the absolute path to the pseudo-labeled dataset root directory and the training split list file. The configuration file looks as follows:

```
{
    "root": "/path/to/data/Pseudo-Labels",
    "train": {
        "pseudo": "./data/pseudo.train"
    }
}
```

B.4 Loss Curves

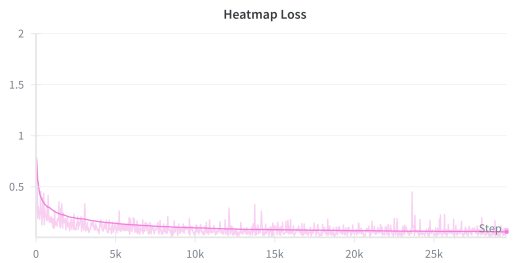
The subsequent graphs show the loss curves of the individual loss terms as they evolved over the 10-epoch pseudo-label fine-tuning run. The loss terms correspond to the same objectives as described in Appendix A.4 for the baseline training.

(a) Width-height loss during refinement



(b) Offset loss during refinement

(c) Identity embedding loss during refinement



(d) Heatmap loss during refinement