Mobility Helps Learning: Unsupervised Model Adaptation for Object Recognition via Movement

Yanan Ma, Yihang Tao, Zhengru Fang, Senkang Hu
Department of Computer Science, City University of Hong Kong
{yananma8-c, yihang.tommy, zhefang4-c, senkang.forest}@my.cityu.edu.hk

Xianhao Chen
Department of Electrical and Computer Engineering
The University of Hong Kong
xchen@eee.hku.hk

Yuguang Fang*
Department of Computer Science
City University of Hong Kong
my.fang@cityu.edu.hk

Abstract

Movable agents, such as autonomous vehicles and robots, should be able to autonomously adapt their pre-trained object recognition models to new environments without human annotations. While existing test-time adaptation (TTA) methods address label-free model adaptation, they typically treat test data as independent snapshots, without constructively exploiting a rich, unique source of supervision: the significant variance in prediction quality as an agent observes the same object from different distances and viewpoints. To capitalize on this, we propose `MoCaFe`, a model-agnostic and hyperparameter-insensitive paradigm that leverages motion-induced prediction variance for unsupervised adaptation. Specifically, we first develop mobility-calibrated filtering of pseudo-labels, which constructs reliable pseudo-labels by fusing cross-view predictions using inverse-variance weights. The filtered pseudo-labels then guide the selection of "hard samples" for effective learning. To further suppress pseudo-label corruption, we introduce `p-MoCaFe`, an optional extension that uses a public dataset to assign weighting factors to data samples, yielding an estimator of the clean loss with bounded bias. Extensive experiments on autonomous driving (nuScenes, KITTI) and embodied-agent datasets show that `MoCaFe` significantly outperforms state-of-the-art TTA schemes, achieving up to approximately 20% accuracy gain.

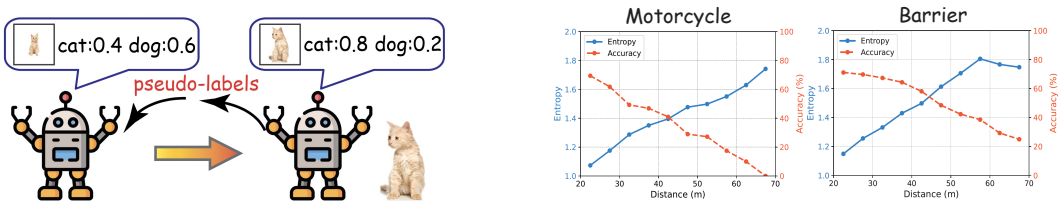
1 Introduction

Object recognition models are notoriously weak at generalization, often suffering significant performance degradation even with minor changes in data distributions [1, 2]. This limitation is particularly critical for mobile agents such as autonomous vehicles and robots, as their very purpose is to navigate and operate in a world that is, by nature, new and unpredictable. To bridge the generalization gap, recognition models often need to continually adapt to new environments [3, 4]. However, traditional supervised adaptation is impractical in this context, since it relies on a labor-intensive and time-consuming manual labeling process [5]. For instance, adapting models locally on a vehicle would require drivers' manual labeling, which is often impractical. This creates a critical need for an automated, unsupervised adaptation framework with minimal human intervention.

To address this need, unsupervised model adaptation, or test-time adaptation (TTA), has emerged as a promising field [6–9]. In the vision domain, existing TTA methods have shown promise,

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Illustration of mobility helps learning.

(b) Accuracy and entropy vs. distance.

Figure 1: (a) The agent is more likely to produce the correct class, i.e., cat, when moving closer to the object. Intuitively, the high-confidence predictions (on the right) can serve as pseudo-labels to guide model adaptation over samples with lower confidence (on the left). (b) Average entropy and classification accuracy versus the distance for two classes of objects, i.e., barriers and motorcycles, in the nuScenes dataset with a pre-trained ResNet-50. Accuracy **varies significantly as agents move**.

often by employing uncertainty-guided pseudo-labeling [10], image/feature generation [11], and contrastive learning [6, 7]. However, the effectiveness of prior TTA methods is still hindered by two shortcomings. ❶ *First*, existing approaches typically treat test data as independent snapshots, without constructively exploiting a rich and freely available source of information: a movable agent observes the same object instance spatiotemporally from varying distances and viewpoints. This motion naturally induces significant variance in prediction quality; an object that is ambiguous from afar is often easily classified up close, and a partially occluded object may become fully visible in subsequent frames (Fig. 1). Consequently, high-quality predictions upon movement can naturally serve as "teacher" outputs[2] without manual labeling. Note that, although leveraging mobility to learn has been exploited in the semi-/unsupervised learning context [12, 13], these schemes are mostly focused on representation learning rather than unsupervised model adaptation, particularly for object recognition. ❷ *Second*, prior methods are often hyperparameter-sensitive or model-specific [14]. A framework meticulously tuned for one environment or architecture may fail or require careful re-tuning when applied to another, rendering it unreliable for real-world deployment. Consequently, two research questions remain underexplored in TTA for object recognition tasks:

- ***Q1:*** *How can we effectively harness movement as a source of supervision for object recognition without manual annotations?*
- ***Q2:*** *How can we design a model-agnostic, hyperparameter-insensitive, and robust adaptation framework that exploits mobility-induced information?*

To answer the research questions above, this paper proposes `MoCaFe`, a novel TTA paradigm for object recognition that leverages motion-induced prediction variance for model adaptation. We treat cross-view predictions as noisy observations of an underlying ground-truth state and fuse them into a mobility-calibrated pseudo-label. A data-sampling process is then employed to select student samples for learning with large cross-entropy relative to the filtered pseudo-label, facilitating effective learning via movement (answering **Q1**). Importantly, the filtered pseudo-label relies on a simple, inverse-variance prediction combination, which is hyperparameter-insensitive, model-agnostic, and robust to model over-confidence (answering **Q2**)[3]. We further develop `p-MoCaFe`, an optional calibrated extension based on a public dataset, and theoretically show that its calibrated objective yields a bounded-bias estimator of the clean loss. Our contributions are summarized as follows.

- We present the idea of "mobility helps learning", a mobility-driven TTA paradigm that exploits the fact that movable agents naturally observe the same object from multiple distances and viewpoints. This enables cross-view self-supervision without requiring manual labels from the target adaptation environment.
- We propose `MoCaFe`, a mobility-calibrated filtering scheme for pseudo-labeling inspired by information filtering. This framework is coupled with a cross-entropy-based data sampling process to selectively learn from informative student samples.

[2] In this paper, we refer to the constructed pseudo-label used for supervising model training as teacher outputs, and the training data samples guided by these outputs as student samples.

[3] Framework-induced hyperparameters exclude common hyperparameters, such as learning rates, which are needed by general learning frameworks.

- We develop `p-MoCaFe`, an optional calibrated extension for `MoCaFe` based on a public dataset. We theoretically show that the calibrated objective yields a bounded-bias estimator of the clean loss.
- We conduct extensive experiments across different model architectures and datasets. The results show that both `MoCaFe` and `p-MoCaFe` significantly improve adaptation performance, with up to approximately 20% accuracy gain for transformer-based models over strong TTA baselines, while maintaining robustness across hyperparameter settings and validation dataset choices.

2 Related Work

Test-Time Adaptation. TTA adapts models to unlabeled data without requiring access to the source dataset [6, 10, 14–18]. In [7], test-time training introduces a self-supervised auxiliary rotation prediction task to be optimized jointly during both source and target training. However, this approach requires modification to pre-training. In [6], Chen *et al.* propose to refine predictions within a self-supervised strategy. In [10], Litrico *et al.* propose a framework that reweights classification loss according to pseudo-label uncertainty to improve robustness by progressively refining the pseudo labels via neighborhood knowledge aggregation. Nevertheless, the aforementioned works treat test data as independent samples. In [19], by observing that data can be temporally correlated in application scenarios, e.g., autonomous driving, Gong *et al.* propose a scheme against such non-IID data streams with temporal correlation. However, while the time correlation of data frames upon agent movement is considered, their goal is to combat its negative effects rather than exploit it as a constructive source for adaptation. Overall, prior TTA methods mainly rely on model-internal supervisory signals on *individual* samples (i.e., pseudo-labeling or entropy minimization) without *constructively* leveraging the cross-view prediction relationships and variances. As a result of their design, most existing TTA approaches remain sensitive to hyperparameter choices or model architecture [20]. Our framework stands out by not only exploiting a novel dimension of design, i.e., harnessing movement as a source of supervision for object recognition, but also delivering superior performance while being model-agnostic and hyperparameter-insensitive.

Other Unsupervised Learning Approaches. Apart from TTA, there are some other unsupervised learning frameworks, yet they are designed for different goals. **Contrastive learning** is a powerful unsupervised learning paradigm that aims to learn an embedding space [21]. A seminal work in this field is InfoNCE [22], which introduces a contrastive loss function to compare positive samples against multiple negatives. Many recent popular frameworks, such as DINO and iBOT, exploit multiple views of the original images (e.g., by view crops or masking) via *data augmentation* for contrastive learning [23–27]. Conceptually similar, Agrawal *et al.* propose a feature learning approach based on egomotion of agents [12]. Nevertheless, these contrastive learning approaches aim to learn transferable features during the *pre-training stage*, which are then applied to specific downstream tasks using labeled data. This fundamentally differs from our approach, which targets the model adaptation (*fine-tuning*) stage for object recognition, and thus, faces unique challenges in pseudo-label generation and label-noise mitigation.

Knowledge distillation (KD) [28] is increasingly applied to unlabeled datasets by employing a powerful teacher model to generate pseudo labels [29]. In [30], a "debiasing" reweighting framework is proposed to reweight the loss function of a student model on an unlabeled dataset. In [31], a student-label mixing (SLaM) approach is proposed to modify the student's loss by incorporating the teacher's noise model. However, our approach differs from these label-free KD mechanisms because both "teacher" and "student" predictions *come from the same pre-trained model*.

3 Problem Definition

We consider a classical multiclass classification problem. Let $\mathcal{X}$ denote the input space and $\mathcal{Y} = \{1, \dots, C\}$ represent the label space with C distinct classes. Our training dataset $\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ consists of n samples drawn from an unknown distribution Π over $\mathcal{X} \times \mathcal{Y}$. The task is to learn a logit predictor $f : \mathcal{X} \to \mathbb{R}^C$, whose predicted class is $\arg\max_c [f(x)]_c$, that minimizes the risk associated with misclassification. To measure the performance of f, we define the expected risk

$$R(f) = \mathbb{E}[\ell(y, f(x))], \tag{1}$$

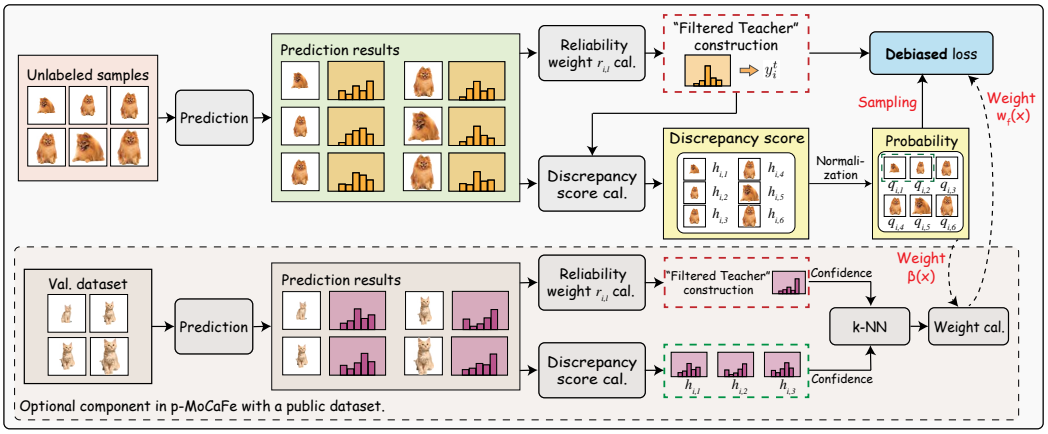


Figure 2: **Overview of the** `MoCaFe` **and** `p-MoCaFe` **frameworks detailed in Section 4.** Multi-view predictions are aggregated via reliability weighting to construct a robust "Filtered Teacher" pseudo-label. Discrepancy scores against this teacher are then used to select the most informative student samples for training. In the p-MoCaFe extension, an additional public dataset is used to calibrate debiasing weights, yielding an unbiased empirical risk for robust model adaptation.

where (x, y) is sampled from Π and $\ell(\cdot)$ is a loss function. We aim to find the function $f^\star$ that minimizes the empirical risk

$$f^\star = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{(x,y)\in S} \ell(y, f(x)), \tag{2}$$

where $\mathcal{F}$ is a predefined hypothesis class of predictors, often parameterized by their weights.

However, as can be seen, the above classical optimization demands ground-truth label y, which is assumed to be unavailable in our case. In what follows, we propose the `MoCaFe` framework to effectively optimize f by exploiting the different spatial-temporal viewpoints on object instances.

4 Proposed `MoCaFe` Framework

In this section, we first elaborate on our fully unsupervised `MoCaFe` framework (Section 4.1), which constructs filtered pseudo-labels by aggregating multiple predictions of the same object according to their estimated observation reliability. We then introduce `p-MoCaFe` (Section 4.2), an optional extension that utilizes a public dataset to achieve a bounded-bias estimator of the clean loss.

4.1 `MoCaFe` Scheme

4.1.1 Reliability-Weighted Pseudo-Label Construction

Let $\mathcal{L}_i = \{x_{i,1}, \ldots, x_{i,L_i}\}$ denote a sequence of $L_i \geq 2$ frames containing the same object i, which can be obtained based on temporal correlation and class-agnostic object-tracking methods. We aim to construct a pseudo-label from each sequence $\mathcal{L}_i$. To theoretically justify our framework, we make the following assumption [32].

Assumption 1. *We assume that the multi-view logit predictions for an object i can be modeled as noisy observations of a latent semantic state. Specifically, the logit vector $\mathbf{z}_{i,l} \in \mathbb{R}^C$ for frame l is formulated as*

$$\mathbf{z}_{i,l} = \mathbf{s}_i + \boldsymbol{\epsilon}_{i,l}, \quad \boldsymbol{\epsilon}_{i,l} \sim \mathcal{N}(\mathbf{b}_{i,l}, \sigma^2(u_{i,l})\mathbf{I}), \tag{3}$$

where $\mathbf{s}_i$ is the true state and $\boldsymbol{\epsilon}_{i,l}$ is stochastic noise. The mean of the noise $\mathbf{b}_{i,l}$ captures prediction bias (e.g., from domain shift or occlusion), while the variance $\sigma^2(u_{i,l})$ is conditioned on an uncertainty score $u_{i,l}$[4].

[4]Here, $u_{i,l}$ is set to the normalized predictive entropy: $u_{i,l} = -\frac{1}{\log C}\sum_{c=1}^{C} p_{i,l,c} \log p_{i,l,c}$ without requiring ground-truth labels. If distance measurements $d_{i,l}$ are available (e.g., via radar), the variance can be extended to $\sigma^2(d_{i,l}, u_{i,l})$.

To estimate observation variance, we partition the uncertainty space into discrete bins defined by $\mathcal{G}_k = \{(i,l) : u_{i,l} \in U_k\}$, where U_k denotes the uncertainty interval[5]. The variance for $\mathcal{G}_k$ is empirically estimated as

$$\hat{\sigma}_k^2 = \frac{1}{|\mathcal{G}_k|} \sum_{(i,l)\in\mathcal{G}_k} \rho_{i,l}, \quad \text{with } \rho_{i,l} = \frac{1}{C} \left\| \mathbf{z}_{i,l} - \widehat{\mathbf{s}}_{i,-l} \right\|_2^2, \tag{4}$$

where $\widehat{\mathbf{s}}_{i,-l} = \frac{1}{L_i - 1} \sum_{\tau \neq l} \mathbf{z}_{i,\tau}$. Here, $\rho_{i,l}$ is the residual inconsistency of frame l based on *leave-one-out* sequence consensus. By mapping this back to individual observations, $\widehat{\sigma}^2(u_{i,l}) = \widehat{\sigma}_k^2$, we define the reliability weight for each frame via inverse-variance weighting

$$r_{i,l} = \frac{1}{\hat{\sigma}^2(u_{i,l}) + \varepsilon_r}, \tag{5}$$

where $\varepsilon_r > 0$ is a small stabilizing constant. Then, we aggregate the weighted logits to construct the filtered pseudo-label vector $\boldsymbol{\mu}_i$, yielding the sequence-level pseudo-label y_i^t

$$\boldsymbol{\mu}_i = \frac{\sum_{l=1}^{L_i} r_{i,l} \mathbf{z}_{i,l}}{\sum_{l=1}^{L_i} r_{i,l}}, \quad y_i^t = \arg\max_{c\in\{1,\ldots,C\}} [\boldsymbol{\mu}_i]_c. \tag{6}$$

Unlike standard confidence-based selection, this formulation relies on a reliability-weighted aggregation of multiple observations. Predictions with lower residual variance are assigned higher weights and contribute more to the filtered pseudo-labels.

Theoretical Analysis. We establish the following theoretical results to justify our framework. For notational simplicity, we drop the object-specific index i and consider a generic sequence length L.

Lemma 1. *The stochastic posterior variance* $\Sigma_L = \left(\sum_{l=1}^{L} \sigma_l^{-2} \right)^{-1}$ *of the filtered prediction is lower than that of any individual observation:*

$$\Sigma_L < \min_{1\le l\le L} \sigma_l^2. \tag{7}$$

Proposition 1. *Assume* $\mathbf{b}_l = \mathbf{0}$ *for all* l*, with Lemma 1, the theoretical misclassification upper bound of the filtered pseudo-label* y^t *is lower than that of the single best prediction* y_{best}*:*

$$\mathbb{P}(y^t \neq c^\star) \le \sum_{c\neq c^\star} \exp\left(-\frac{\Delta_{0,c}^2}{4\Sigma_L} \right) < \sum_{c\neq c^\star} \exp\left(-\frac{\Delta_{0,c}^2}{4\sigma_{best}^2} \right), \tag{8}$$

where $c^\star$ *is the ground-truth class, and* y_{best} *denotes the prediction from the single best observation corresponding to the minimum variance* $\sigma_{best}^2 = \min_{1\le l\le L} \sigma_l^2$*.*

The detailed proofs for Lemma 1 and Proposition 1 are provided in Appendix A.1 and A.2.

Remark 1. *Proposition 1 demonstrates that the upper bound of the filtered teacher is smaller than the upper bound of any single "best" prediction, assuming bias* $\mathbf{b}_l = \mathbf{0}$ *(i.e., the prediction is the ground-truth prediction with zero-mean Gaussian noise, as in [32]). While this assumption may not always hold, it provides theoretical justification for the design of the filtered pseudo-labels. Empirically, our experiments show that the filtered pseudo-label is more robust than simply selecting the highest-confidence prediction from each sequence as the teacher (see Appendix B.1).*

4.1.2 Informative Sampling with Debiasing

To facilitate efficient adaptation, we preferentially sample informative frames that "disagree" with the filtered pseudo-label, which are essentially "hard examples". Specifically, we define a discrepancy score between the filtered pseudo-label and the individual prediction $h_{i,l} = \ell(y_i^t, f(x_{i,l}))$, where ℓ denotes the loss (e.g., cross-entropy). A larger $h_{i,l}$ indicates high inconsistency, suggesting that the frame should be selected for adaptation. We then construct sampling probabilities

$$q_{i,l} = \frac{h_{i,l} + \eta}{\sum_{i'=1}^{N} \sum_{\tau=1}^{L_{i'}} (h_{i',\tau} + \eta)}, \ \ \forall i, \forall l, \tag{9}$$

[5]With available distance measurements, this can be extended to a 2D mobility-uncertainty partition: $\mathcal{G}_k = \{(i,l) : d_{i,l} \in D_j, u_{i,l} \in U_k\}$, where D_j is the distance interval. Since distance is solely used for interval assignment, coarse range estimates (e.g., 0–40 m, 40–80 m, >80 m) are sufficient.

Algorithm 1: The MoCaFe Framework

Input: Training dataset $\mathcal{X}$ composed of tracked sequences $\{\mathcal{L}_i\}$, pre-trained model f
Output: Adapted model $f^\star$

for *each sequence* $\mathcal{L}_i \in \mathcal{X}$ **do**
  for *each frame* $x_{i,l} \in \mathcal{L}_i$ **do**
    Compute logits $\mathbf{z}_{i,l} = f(x_{i,l})$ and uncertainty $u_{i,l}$;
  end
  Compute $\rho_{i,l}$ for all frames in $\mathcal{L}_i$ via Eq. (4);
end
Partition all observations $\{(i,l)\}$ into uncertainty bins $\{\mathcal{G}_k\}_{k=1}^K$ based on $u_{i,l} \in U_k$;
for *each bin* $\mathcal{G}_k$ **do**
  Compute pooled empirical variance $\widehat{\sigma}_k^2 = \frac{1}{|\mathcal{G}_k|}\sum_{(i,l)\in\mathcal{G}_k}\rho_{i,l}$ via Eq. (4);
  Compute bin reliability weight $r_k = \frac{1}{\widehat{\sigma}_k^2+\varepsilon_r}$ via Eq. (5);
end
Assign frame reliability $r_{i,l} \leftarrow r_k$ for each $(i,l) \in \mathcal{G}_k$;
for *each sequence* $\mathcal{L}_i \in \mathcal{X}$ **do**
  Construct $\boldsymbol{\mu}_i$ and y_i^t via Eq. (6) and assign $y^t(x_{i,l}) \leftarrow y_i^t$ and $\boldsymbol{\mu}(x_{i,l}) \leftarrow \boldsymbol{\mu}_i$;
end
$\mathcal{Y}^t \leftarrow \{y^t(x) \mid x \in \mathcal{X}\}$;
Calculate $h(x) = \ell(y^t(x), f(x))$ for all $x \in \mathcal{X}$;
Compute $q(x) \leftarrow \frac{h(x)+\eta}{\sum_{x'\in\mathcal{X}}(h(x')+\eta)}$ for all $x \in \mathcal{X}$; Sample $\mathcal{X}_m \subset \mathcal{X}$ according to $q(x)$;
Adapt model f over $\mathcal{X}_m$ by minimizing $\hat{R}_q(f)$ via Eq. (10);

where $\eta > 0$ is a small value preventing zero-probability sampling for frames already aligned with pseudo-labels. Accordingly, we select m samples from the dataset, forming the selected dataset $\mathcal{X}_m$.

Since the sampling process above draws from an instrumental distribution $\mathbb{Q}$, it may not perfectly match the true data distribution Π, potentially introducing bias. Assuming the filtered pseudo-label is perfect, this empirical risk can be easily debiased by weighting factors $\beta(x_{i,l})$ via

$$\hat{R}_q(f) = \frac{1}{m}\sum_{x_{i,l}\in\mathcal{X}_m}\beta(x_{i,l})\ell(y_i^t, f(x_{i,l})). \tag{10}$$

Assuming $y^t(x)$ is the ground-truth label, i.e., $y^t(x) = y$, we have the following proposition.

Proposition 2. *If* $\beta(x) = \pi(x)/q(x)$*, where* $\pi(x)$ *is the probability density of the distribution* Π *and* $q(x)$ *is the probability density of the instrumental distribution* $\mathbb{Q}$*, then* $\hat{R}_q(f)$ *in* (10) *is an unbiased estimator of* $R(f)$*.*

Note that $\beta(x) \propto 1/q(x)$ if x is drawn i.i.d. from Π. Proposition 2 is a direct consequence of the definition of $\beta(x)$, as shown in [33]. The overall MoCaFe framework is illustrated in Fig. 2 and summarized in Alg. 1.

4.2 p-MoCaFe: An Optional Extension Using Public Datasets

The MoCaFe framework yields an unbiased estimator of $R(f)$ only if the filtered pseudo-labels $y^t(x)$ constitute the ground truth. Since it inevitably contains inaccuracies, we introduce p-MoCaFe, an optional extension using a public dataset to debias the empirical risk via data weighting. In particular, after the whole process of MoCaFe, p-MoCaFe replaces weight $\beta(x)$ in (10) with $w_f(x)$

$$w_f(x) = \frac{\beta(x)}{1 + p(x)\,(d_f(x) - 1)}, \quad \text{with } d_f(x) = \frac{\ell\left(f_{\text{false}}(x), f(x)\right)}{\ell\left(y, f(x)\right)}. \tag{11}$$

The weighted empirical risk, therefore, can be expressed as

$$R^w(f) = \frac{1}{m}\sum_{x\in\mathcal{X}_m} w_f(x)\ell(y^t(x), f(x)). \tag{12}$$

From Eq. (11), we can observe that the weight for each sample x depends on two factors: *(i) the likelihood that the teacher corrupts its label, and (ii) the extent to which this corrupted label distorts the risk.* Specifically, if the probability of a corrupted label is high or the distortion term is large, we should assign a smaller weight to minimize the influence of the teacher's erroneous prediction (See Appendix A.4). We further show that the weighted empirical risk $R^w(f)$ in Eq. (12) is an unbiased estimation of $R(f)$.

Proposition 3. *Assume that $\mathcal{X}_m$ is drawn from the induced distribution $\tilde{\mathbb{Q}}_t$, then we have:*

$$\mathbb{E}\left[R^w(f)\right] = R(f). \tag{13}$$

The detailed proof follows from [30], which can be found in Appendix A.5. Given a sufficiently large dataset sampled from $\tilde{\mathbb{Q}}_t$, optimizing an unbiased estimator for the risk should provide a better approximation for $\min_{f\in\mathcal{F}} R(f)$ than optimizing an estimator with constant bias.

Estimating Weights (11). To calculate the weights in Eq. (11), one hurdle is that $p(x)$ and $d_f(x)$ are usually unknown. To address this issue, we assume the existence of a public dataset $\mathcal{S}_v = \{(x_1, y_1), (x_2, y_2), \ldots, (x_v, y_v)\}$. We apply the multi-view filtering process to the validation sequences to construct pseudo-labels $y^t(x)$, and the remaining frames are designated as student images. We then estimate the weights $\hat{p}(x)$ and $\hat{d}_f(x)$ via the k-Nearest Neighbors (k-NN) method (see Appendix A.6 for details). Since the k-NN approach introduces approximation errors, in what follows, we generalize Proposition 3 and the results in [30] by establishing that the deviation of our weighted empirical risk from the true risk remains bounded by these estimation errors.

Proposition 4. *The weighted empirical risk estimator, given by $\hat{R}^{\tilde{w}}(f) = \frac{1}{m}\sum_{x\in\mathcal{X}_m} \tilde{w}_f(x)\ell(y^t(x), f(x))$, satisfies:*

$$\begin{aligned}\left|\mathbb{E}[\hat{R}^{\tilde{w}}(f)] - R(f)\right| \leq\ & \mathbb{E}_{x\sim\tilde{\mathbb{Q}}_t}\Big[\tilde{w}_f(x)\ell(y, f(x))\big|p(x)(d_f(x)-1) - \hat{p}(x)(\hat{d}_f(x)-1)\big|\Big] \\ & + \mathbb{E}_{x\sim\tilde{\mathbb{Q}}_t}\Big[\big|\tilde{w}_f(x) - \hat{w}_f(x)\big|\big[1+\hat{p}(x)(\hat{d}_f(x)-1)\big]\ell(y, f(x))\Big],\end{aligned} \tag{14}$$

where $\hat{w}_f(x)$, $\hat{p}(x)$, and $\hat{d}_f(x)$ denote the raw estimates via k-NN, and $\tilde{w}_f(x) = \min\{\hat{w}_f(x), 1\}$ denotes the clipped weight.

The detailed proof is provided in Appendix A.7. Proposition 4 reveals that the risk of `p-MoCaFe` framework contains two parts: estimation bias (the first term), which scales proportionally with the k-NN approximation error, and clipping bias (the second term), which arises when the raw estimated weight is truncated by a boundary, e.g., 1. Notably, we show in Appendix B.1 that `p-MoCaFe` consistently outperforms `MoCaFe` even with a substantial domain gap between the public and target datasets. The `p-MoCaFe` framework is summarized in Alg. 2 in Appendix A.3.

5 Experiments

5.1 Experimental Settings

Dataset. We conduct experiments using autonomous driving datasets (nuScenes [34] and KITTI [35]), as well as embodied agent datasets (selected from JRDB [36], MOTS [37], and LaSOT [38]). Since these datasets are captured by movable agents in the real world, they naturally provide images of the same object from varying distances/viewpoints and with different levels of confidence and accuracy. We construct instance sequences by extracting the same objects across images using ground-truth associations to demonstrate the theoretical and empirical potential of our method. For practical evaluation, we also report performance using an off-the-shelf class-agnostic object tracker. For all experiments, we initially pre-train these models in Singapore Onenorth on the nuScenes dataset. We test i) **cross-region** adaptation, which adapts the pre-trained model in the Boston Seaport scenario on the nuScenes dataset, ii) **cross-dataset** adaptation, which adapts the model on the KITTI dataset, and iii) **cross-domain** adaptation, which adapts the model on the embodied agent dataset. For target datasets with different label spaces, the classification head is adjusted. The latter two are provided in Appendix B.4 and B.5, which consistently demonstrate our schemes outperform other baselines.

Implementation Details. We adopt two widely used model architectures: a convolutional neural network (ResNet-50) and a transformer (Swin-T) [39]. We also demonstrate that our method can be

Table 1: Performance in the Boston Seaport scenario on the nuScenes dataset under varying pre-trained model accuracy (i.e., 75%, 70%, and 65%). The value in the bracket represents the standard deviation. **Boldface** and Underline highlight the highest and second-highest accuracy, except for the FSL benchmark. ↑ and ↓ indicate performance improvement/degradation.

	ResNet-50			**Swin-T**		
Pre-trained	75%	70%	65%	75%	70%	65%
w/o MoCaFe	76.18↑(± 0.28)	72.05↑(± 0.28)	62.17↓(± 0.37)	76.69↑(± 0.26)	72.55↑(± 0.25)	40.23↓(± 0.43)
AdaContrast	77.88↑(± 0.19)	73.63↑(± 0.15)	66.51↑(± 0.18)	76.85↑(± 0.33)	72.83↑(± 0.47)	62.28↓(± 0.61)
PLUE	77.94↑(± 0.12)	73.82↑(± 0.21)	67.28↑(± 0.29)	77.43↑(± 0.24)	73.55↑(± 0.30)	59.91↓(± 0.35)
RoTTA	78.67↑(± 0.21)	75.13↑(± 0.33)	68.20↑(± 0.31)	—	—	—
C-SFDA	78.71↑(± 0.22)	75.37↑(± 0.26)	68.34↑(± 0.34)	77.75↑(± 0.18)	72.20↑(± 0.19)	66.43↑(± 0.25)
UPA	76.87↑(± 0.36)	74.13↑(± 0.34)	57.60↓(± 0.48)	78.24↑(± 0.26)	73.15↑(± 0.23)	68.12↑(± 0.28)
AnCon	79.23↑(± 0.29)	75.61↑(± 0.36)	68.23↑(± 0.38)	80.38↑(± 0.27)	76.28↑(± 0.30)	70.59↑(± 0.32)
MoCaFe	80.81↑(± 0.23)	77.25↑(± 0.25)	69.32↑(± 0.24)	89.60↑(± 0.11)	89.34↑(± 0.15)	89.27↑(± 0.16)
p-MoCaFe	**81.23↑(± 0.13)**	**77.93↑(± 0.16)**	**69.74↑(± 0.25)**	**90.19↑(± 0.17)**	**90.07↑(± 0.18)**	**90.02↑(± 0.12)**
FSL	88.66↑(± 0.12)	87.83↑(± 0.18)	86.87↑(± 0.25)	92.30↑(± 0.06)	92.12↑(± 0.15)	91.98↑(± 0.28)

extended to adapt the classification part of Faster R-CNN, a well-known object detection model. We first perform inference over the unlabeled target sequences using the source-pretrained checkpoint to obtain the predictions (logits). Note that the varying pre-trained model accuracies (i.e., 75%, 70%, and 65%) are evaluated directly on this target dataset. We then fine-tune the model over $\mathcal{X}_m$ using the weighted objective in Eq. (10) for `MoCaFe` and Eq. (12) for `p-MoCaFe`. The adaptation process runs for 50 epochs for ResNet-50 and Swin-T, and 200 epochs for Faster R-CNN. For ResNet-50, the learning rate is 10^{-4} for pre-training, 10^{-5} for `MoCaFe` adaptation and 2×10^{-5} for `p-MoCaFe` adaptation, a momentum coefficient of 0.9, and a weight decay of 5×10^{-4}. The Swin-T configuration is adopted with a window size of 7 and a patch size of 4. The base learning rate is set to 10^{-5} with a 5-epoch warmup. We use the Singapore Onenorth scenario in the nuScenes dataset as the public dataset, with the size of $v = 8000$. For hyperparameter k induced by `p-MoCaFe`, we set $k = \lceil \sqrt{v}/2 \rceil$ and provide ablation analysis in Section 5.4.

5.2 Comparison with Benchmarks

Baselines. We compare our proposed `MoCaFe` and `p-MoCaFe` schemes with the following benchmarks: **AdaContrast** [6], **PLUE** [10], **RoTTA** [14], **C-SFDA** [15], **UPA** [16], and **AnCon** [17]. We also include two benchmarks: **w/o MoCaFe**, a baseline employing uniform weighting and random sampling, and **FSL** (Fully-Supervised Learning), which serves as the upper bound. Since the aforementioned TTA benchmarks are released with ResNet-based implementations, their default hyperparameter settings are not directly applicable to transformers, highlighting their sensitivity to hyperparameters across different model architectures. We therefore reimplement them on Swin-T with appropriate hyperparameter choice. Moreover, we exclude RoTTA for Swin-T, as its reliance on robust batch normalization is incompatible with transformer architectures.

Baseline Comparison. Table 1 summarizes the adaptation performance on the Boston Seaport scenario using both Swin-T and ResNet-50 models. As observed, the proposed `MoCaFe`/`p-MoCaFe` framework consistently outperforms other benchmarks significantly. Our method demonstrates particularly significant improvements on Swin-T. We attribute this to Swin-T's strong representation and adaptation capabilities [40], but these capabilities can only be exploited when the supervisory signal is sufficiently reliable. Conventional methods construct adaptation signals from individual-frame confidence, entropy, or neighborhood predictions. When these frame-level signals are corrupted, they may reinforce incorrect predictions. Instead, `MoCaFe` fuses multiple views of the same object via an information filter to construct a more reliable sequence-level pseudo-label. Moreover, although some baselines employ their own weighting or sample-selection mechanisms, none leverage cross-view inverse-variance fusion and the discrepancy-driven sampling strategy. Besides, `MoCaFe` achieves performance comparable to the `p-MoCaFe` framework and surpasses all other baselines across both models. This phenomenon indicates that our approach can achieve superior performance with the `MoCaFe` scheme without requiring a public dataset.

In addition, we have included **per-class accuracy** to further demonstrate the effectiveness of our framework across diverse object categories. As shown in Table 2, our `p-MoCaFe` framework substantially outperforms pre-trained models in terms of per-class accuracy, for ResNet-50 with 9 classes improved and only 1 class degraded and for Swin-T with all 10 classes improved. This is because

Table 2: Per-class test accuracy in the Boston Seaport scenario on the nuScenes dataset. Pre-trained accuracy for the Boston scenario is 70%. $\rightarrow$ represents that the performance remains the same with the pre-trained model.

					ResNet-50					
Class	Car	Pedestrian	Traffic-Cone	Trailer	Bus	Motorcycle	Barrier	Con-Vehicle	Bicycle	Truck
Pre-trained	82.42	77.35	70.46	77.22	62.86	61.11	53.09	49.64	48.15	23.28
w/o MoCaFe	87.41↑	81.05↑	76.95↑	75.32↓	61.90↓	47.22↓	52.31↓	49.68↑	46.91↓	18.84↓
AdaContrast	88.38↑	85.29↑	77.42↑	**81.00**↑	68.10↑	50.05↓	53.09→	40.53↓	37.50↓	14.64↓
PLUE	88.79↑	84.98↑	75.55↑	78.67↑	64.87↑	55.89↓	54.24↑	46.15↓	39.71↓	15.51↓
RoTTA	87.41↑	83.78↑	73.12↑	74.74↓	65.39↑	52.53↓	**58.33**↑	40.53↓	37.62↓	12.87↓
C-SFDA	87.96↑	84.05↑	78.24↑	72.50↓	62.86→	**62.58**↑	54.73↑	51.16↑	35.67↓	18.50↓
UPA	86.43↑	83.02↑	77.84↑	74.97↓	69.52↑	57.67↓	48.84↓	45.86↓	28.88↓	16.73↓
AnCon	88.91↑	86.31↑	81.42↑	78.67↑	68.10↑	51.48↓	54.24↑	47.22↓	42.54↓	16.73↓
MoCaFe	90.34↑	88.68↑	**83.85**↑	76.48↓	71.50↑	52.81↓	54.31↑	50.50↑	53.14↑	19.17↓
p-MoCaFe	**91.19**↑	**89.77**↑	83.78↑	77.90↑	**73.38**↑	61.18↑	56.85↑	**56.21**↑	**54.42**↑	**20.71**↓
					Swin-T					
Class	Car	Pedestrian	Traffic-Cone	Trailer	Bus	Motorcycle	Barrier	Con-Vehicle	Bicycle	Truck
Pre-trained	85.45	73.42	74.68	72.78	60.95	65.56	52.25	51.76	49.38	24.37
w/o MoCaFe	87.41↑	78.24↑	76.05↑	76.53↑	70.47↑	75.56↑	56.32↑	49.92↓	46.58↓	19.14↓
AdaContrast	89.72↑	80.02↑	78.10↑	73.90↑	66.83↑	73.92↑	50.37↓	45.96↓	48.65↓	15.13↓
PLUE	90.08↑	78.10↑	78.26↑	76.53↑	64.43↑	68.37↑	54.18↑	47.31↓	49.19↓	16.73↓
C-SFDA	90.30↑	81.23↑	77.41↑	74.55↑	65.95↑	66.83↑	55.83↑	45.66↓	36.59↓	18.96↓
UPA	88.90↑	80.20↑	80.61↑	71.65↓	69.48↑	58.23↓	58.59↑	46.46↓	38.90↓	17.27↓
AnCon	92.80↑	84.81↑	82.47↑	75.83↑	67.16↑	62.64↓	52.68↑	43.26↓	39.18↓	17.36↓
MoCaFe	93.73↑	**91.74**↑	94.00↑	90.75↑	**90.69**↑	87.16↑	85.42↑	86.23↑	**76.12**↑	35.60↑
p-MoCaFe	**94.20**↑	91.67↑	**94.03**↑	**92.74**↑	90.13↑	**88.31**↑	**86.88**↑	**87.19**↑	75.41↑	**40.60**↑

`p-MoCaFe` is designed to mitigate the impact of noisy pseudo labels: Eq. (11) shows that when the probability of pseudo-label corruption is high, and the distortion is large, a smaller weighting factor $w_f(x)$ shall be assigned to the sample. The simulation results demonstrate that our framework substantially outperforms baselines in combating per-class biases.

5.3 Impact of Various Key Factors

Effect of Pre-trained Model Accuracy. In Table 1, we also observe that adaptation performance improvement/decline is greatly influenced by pre-trained model accuracy. A lower-accuracy model produces more corrupted pseudo labels, which thereby misguide the adaptation process and even cause severe degradation, for example, with Swin-T (65% pre-trained accuracy), "w/o MoCaFe" drops accuracy by over 20%. However, our `MoCaFe` framework demonstrates consistent and remarkable performance improvement even in such challenging scenarios.

Effect of Sequence Construction. Our framework only requires a weak assumption that the same objects can be associated across frames, which can be achieved by class-agnostic tracking based on off-the-shelf object-tracking techniques. To validate our argument, we provide simulation results in Table 3 to demonstrate the impact of tracking algorithms. The sequences are constructed using the object tracking algorithm (the average multi-object tracking accuracy (AMOTA) in the Boston Seaport scenario is 67.3%) [41]. Results indicate that off-the-shelf tracking algorithms are sufficient for sequence construction, with only negligible degradation (approximately 0.5%) in performance compared to the ideal, ground-truth-based object associations, e.g., 81.23% to 80.97% with ResNet-50 and 90.19% to 89.81% with Swin-T, demonstrating the practicality of the proposed framework.

Effect of the Number of Selected Samples. Table 11 in Appendix B.3 illustrates the test accuracy across varying numbers of selected training samples, m. The results show that our `MoCaFe` and `p-MoCaFe` frameworks substantially outperform other benchmarks under all evaluated values of m. Notably, even when the training data is reduced to $16,000$, `p-MoCaFe` maintains a robust accuracy of 89.48% on Swin-T and 80.42% on ResNet-50, demonstrating the data efficiency of our approach.

5.4 Ablation Study

We conduct comprehensive ablation studies with detailed results and tables provided in Appendix B.1. Our principal findings can be summarized as follows. **Robustness of pseudo-label construction**: In comparison to selecting the highest-confidence (lowest entropy) prediction from each sequence as the teacher (i.e., the max-confidence teacher), the filtered teacher in `MoCaFe` demonstrates stronger robustness. Specifically, the filtered teacher clearly surpasses the max-confidence teacher when the pre-trained models are very poor (around 50% accuracy), a regime in which learning from the max-

Table 3: Performance in the Boston Seaport scenario on the nuScenes dataset with sequences constructed by an off-the-shelf object tracking algorithm.

	ResNet-50			**Swin-T**		
Pre-trained	75%	70%	65%	75%	70%	65%
w/o MoCaFe	75.53↑(± 0.24)	71.41↑(± 0.26)	61.48↓(± 0.39)	76.10↑(± 0.25)	71.96↑(± 0.27)	39.51↓(± 0.39)
AdaContrast	76.45↑(± 0.14)	72.72↑(± 0.18)	65.46↑(± 0.19)	76.23↑(± 0.26)	72.16↑(± 0.22)	60.39↓(± 0.54)
PLUE	77.16↑(± 0.16)	73.04↑(± 0.25)	66.80↑(± 0.26)	76.88↑(± 0.25)	73.00↑(± 0.32)	57.58 ↓(± 0.34)
RoTTA	77.35↑(± 0.24)	73.85↑(± 0.23)	67.09↑(± 0.27)	—	—	—
C-SFDA	78.03↑(± 0.21)	73.99↑(± 0.28)	67.60↑(± 0.31)	77.11↑(± 0.20)	71.59↑(± 0.23)	65.42↑(± 0.27)
UPA	76.15↑(± 0.30)	73.31↑(± 0.31)	56.81↓(± 0.39)	77.44↑(± 0.28)	72.61↑(± 0.27)	67.55↑(± 0.23)
AnCon	78.88↑(± 0.24)	75.04↑(± 0.33)	67.93↑(± 0.32)	79.73↑(± 0.22)	75.68↑(± 0.27)	69.88↑(± 0.35)
MoCaFe	80.50↑(± 0.22)	76.73↑(± 0.25)	69.01↑(± 0.26)	89.14↑(± 0.19)	88.87↑(± 0.21)	88.78↑(± 0.27)
p-MoCaFe	**80.97↑(± 0.19)**	**77.56↑(± 0.21)**	**69.46↑(± 0.25)**	**89.81↑(± 0.26)**	**89.62↑(± 0.19)**	**89.59↑(± 0.25)**
FSL	88.34↑(± 0.13)	87.40↑(± 0.20)	86.51↑(± 0.28)	91.78↑(± 0.10)	91.68↑(± 0.16)	91.55↑(± 0.25)

confidence teacher (which can be confidently wrong) may mislead model adaptation. **Robustness to public datasets:** `p-MoCaFe` is highly insensitive to domain mismatch and the size of the public dataset used for calibration, exhibiting less than a 0.7% performance drop even when the public dataset size is reduced to 10%. **Hyperparameter stability:** `p-MoCaFe` remains remarkably stable across different neighborhood sizes (k) in k-NN estimation. Note that `MoCaFe` does not introduce any new hyperparameters, and `p-MoCaFe` only has one hyperparameter k induced by our design. Thus, the robustness to k reveals that our framework is hyperparameter-insensitive.

5.5 Extension to Classification in Detection Models.

While this work primarily focuses on single-label classification as a proof-of-concept study for "mobility helps learning," akin to most TTA schemes [6, 10, 16, 17, 19], in practice, modern vehicles, robots, or drones typically implement object detection models to detect and classify *multiple* objects simultaneously. Our `MoCaFe` framework seamlessly extends to adapt the classification modules of such models (detailed in Appendix B.2). Using Faster R-CNN [42, 43] as an example, we adapt only its classification module to the Boston Seaport scenario. As detailed in Appendix B.2, `p-MoCaFe` achieves the highest test accuracy across varying pre-trained model capabilities and sample sizes, with both variants outperforming the baseline by a substantial margin (e.g., $\sim 9\%$).

6 Conclusion

In this paper, we have proposed a general framework for movable agents to adapt object recognition models without manual labeling. Our idea is to exploit the variance in prediction quality for the same object as the agent moves. Since incorrect object predictions can mislead model adaptation, we have developed a `MoCaFe` framework to judiciously sample training data and assign weightings, and to reduce stochastic logit variance via multi-view filtering. We also proposed an optional extension, `p-MoCaFe`, to mitigate the impact of pseudo-label corruption and have theoretically shown that the framework yields unbiased learning with respect to the loss function. Experimental results have demonstrated that our proposed adaptation method can significantly improve model accuracy in object classification tasks and outperform existing test-time adaptation schemes.

Limitations and Future Work. This work serves as a proof-of-concept study for "mobility helps learning", focusing on object classification. However, it is promising to extend this idea to other vision tasks, such as object detection and segmentation, because agent movement can provide valuable information for model adaptation in these tasks as well. These aspects can be exploited in future work.

Acknowledgments.

This work was supported in part by the JC STEM Lab of Smart City, funded by The Hong Kong Jockey Club Charities Trust under Contract 2023-0108; the Research Grants Council of the Hong Kong SAR, China (Project No. CityU 11216324); the Hong Kong SAR Government under the Global STEM Professorship; and the Research Grants Council of Hong Kong under Grant 27213824, Grant 17207826, and Grant CRS HKU702/24.

References

[1] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

[2] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pages 5637–5664. PMLR, 2021.

[3] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):211–229, 1959.

[4] Ligeng Zhu, Lanxiang Hu, Ji Lin, Wei-Ming Chen, Wei-Chen Wang, Chuang Gan, and Song Han. Pockengine: Sparse and efficient fine-tuning in a pocket. In *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 1381–1394, 2023.

[5] Yanan Ma, Senkang Hu, Zhengru Fang, Yun Ji, Yiqin Deng, and Yuguang Fang. Sense4FL: Vehicular crowdsensing enhanced federated learning for object detection in autonomous driving. *IEEE Transactions on Mobile Computing*, 25(8):13004–13018, 2026. doi: 10.1109/TMC.2026.3674333.

[6] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022.

[7] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020.

[8] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4544–4553, 2020.

[9] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020.

[10] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. Guiding pseudo-labels with uncertainty estimation for source-free unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7640–7650, June 2023.

[11] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

[12] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision*, pages 37–45, 2015.

[13] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017.

[14] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932, 2023.

[15] Nazmul Karim, Niluthpol Chowdhury Mithun, Abhinav Rajvanshi, Han-pang Chiu, Supun Samarasekera, and Nazanin Rahnavard. C-sfda: A curriculum learning aided self-training framework for efficient source free domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24120–24131, 2023.

[16] Xi Chen, Haosen Yang, Huicong Zhang, Hongxun Yao, and Xiatian Zhu. Uncertainty-aware pseudo-label filtering for source-free unsupervised domain adaptation. *Neurocomputing*, 575: 127190, 2024.

[17] Taejong Joo and Diego Klabjan. Improving self-training under distribution shifts via anchored confidence with theoretical guarantees. *Advances in Neural Information Processing Systems*, 37:15246–15271, 2024.

[18] Yanan Ma, Yihang Tao, Zhengru Fang, Zihan Fang, Yiqin Deng, Xianhao Chen, and Yuguang Fang. Learning from distributed eyes: Leveraging collaborative perception for automated model adaptation. *arXiv preprint arXiv:2609.18511*, 2026.

[19] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

[20] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8353, 2022.

[21] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL `https://arxiv.org/abs/1807.03748`.

[23] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[24] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

[25] 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. *Transactions on Machine Learning Research Journal*, 2024.

[26] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views, 2019. URL `https://arxiv.org/abs/1906.00910`.

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[28] Geoffrey Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[29] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[30] Fotis Iliopoulos, Vasilis Kontonis, Cenk Baykal, Gaurav Menghani, Khoa Trinh, and Erik Vee. Weighted distillation with unlabeled examples. *Advances in Neural Information Processing Systems*, 35:7024–7037, 2022.

[31] Vasilis Kontonis, Fotis Iliopoulos, Khoa Trinh, Cenk Baykal, Gaurav Menghani, and Erik Vee. SLaM: Student-label mixing for distillation with unlabeled examples. *Advances in Neural Information Processing Systems*, 36, 2024.

[32] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[33] Takafumi Kanamori and Hidetoshi Shimodaira. Active learning algorithm using the maximum weighted log-likelihood estimator. *Journal of statistical planning and inference*, 116(1):149–162, 2003.

[34] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

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

[36] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):6748–6765, 2021.

[37] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 7942–7951, 2019.

[38] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5374–5383, 2019.

[39] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022.

[40] Donghyun Kim, Kaihong Wang, Stan Sclaroff, and Kate Saenko. A broad study of pre-training for domain generalization and adaptation. In *European conference on computer vision*, pages 621–638. Springer, 2022.

[41] Zheng Qin, Le Wang, Sanping Zhou, Panpan Fu, Gang Hua, and Wei Tang. Towards generalizable multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19004, 2024.

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

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

[44] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[45] Trevor Hastie. The elements of statistical learning: data mining, inference, and prediction, 2009.

A Technical Appendices and Supplementary Material

A.1 Proof of Lemma 1

For simplicity of notation in this theoretical analysis, we drop the object index i and consider a generic tracklet of length $L \geq 2$. Under ideal inverse-variance weighting, the reliability weights are $r_l = 1/\sigma_l^2$. Substituting the observation model from Assumption 1 into the expression for the filtered prediction gives:

$$\boldsymbol{\mu}_L = \frac{\sum_{l=1}^{L} r_l(\mathbf{s} + \boldsymbol{\epsilon}_l)}{\sum_{l=1}^{L} r_l}. \tag{15}$$

Since the latent clean semantic state $\mathbf{s}$ is invariant across all frames of the tracked object, it can be factored out:

$$\boldsymbol{\mu}_L = \mathbf{s} + \frac{\sum_{l=1}^{L} r_l \boldsymbol{\epsilon}_l}{\sum_{l=1}^{L} r_l}. \tag{16}$$

Because the noise terms $\{\boldsymbol{\epsilon}_l\}_{l=1}^{L}$ are independent Gaussian random vectors, their affine combination remains Gaussian. Furthermore, since their expected value is given by $\mathbb{E}[\boldsymbol{\epsilon}_l] = \mathbf{b}_l$, the expectation of the filtered prediction evaluates to:

$$\mathbb{E}[\boldsymbol{\mu}_L] = \mathbf{s} + \frac{\sum_{l=1}^{L} r_l \mathbf{b}_l}{\sum_{l=1}^{L} r_l} = \mathbf{s} + \bar{\mathbf{b}}, \tag{17}$$

where $\bar{\mathbf{b}} = \sum_{l=1}^{L} \alpha_l \mathbf{b}_l$ is the weighted systematic bias, with normalized weights $\alpha_l = \frac{r_l}{\sum_{\tau=1}^{L} r_\tau}$.

For the covariance, the independence of the noise terms dictates that cross-covariance terms vanish. Thus, the covariance matrix of $\boldsymbol{\mu}_L$ is given by:

$$\text{Cov}(\boldsymbol{\mu}_L) = \text{Cov}\left(\frac{\sum_{l=1}^{L} r_l \boldsymbol{\epsilon}_l}{\sum_{l=1}^{L} r_l}\right) = \frac{\sum_{l=1}^{L} r_l^2 \text{Cov}(\boldsymbol{\epsilon}_l)}{\left(\sum_{l=1}^{L} r_l\right)^2}. \tag{18}$$

Substituting $\text{Cov}(\boldsymbol{\epsilon}_l) = \sigma_l^2 \mathbf{I} = \frac{1}{r_l}\mathbf{I}$, the numerator simplifies:

$$\text{Cov}(\boldsymbol{\mu}_L) = \frac{\sum_{l=1}^{L} r_l^2 \left(\frac{1}{r_l}\mathbf{I}\right)}{\left(\sum_{l=1}^{L} r_l\right)^2} = \frac{\sum_{l=1}^{L} r_l}{\left(\sum_{l=1}^{L} r_l\right)^2}\mathbf{I} = \left(\sum_{l=1}^{L} r_l\right)^{-1}\mathbf{I} = \Sigma_L \mathbf{I}, \tag{19}$$

where $\Sigma_L = \left(\sum_{l=1}^{L} r_l\right)^{-1} = \left(\sum_{l=1}^{L} \frac{1}{\sigma_l^2}\right)^{-1}$. Thus, we have shown that $\boldsymbol{\mu}_L \sim \mathcal{N}(\mathbf{s} + \bar{\mathbf{b}}, \Sigma_L \mathbf{I})$.

Finally, because all observation variances are positive ($\sigma_l^2 > 0$), it trivially holds that for any sequence of length $L \geq 2$:

$$\sum_{l=1}^{L} \frac{1}{\sigma_l^2} > \max_{1 \leq l \leq L} \frac{1}{\sigma_l^2}. \tag{20}$$

Taking the reciprocal of both sides reverses the inequality, yielding

$$\Sigma_L < \min_{1 \leq l \leq L} \sigma_l^2, \tag{21}$$

which completes the proof.

A.2 Proof of Proposition 1

The filtered teacher misclassifies an observation if there exists at least one incorrect class $c \neq c^\star$ that achieves a higher fused score than the ground-truth class $c^\star$. The probability of this error is given by:

$$\mathbb{P}(y^t \neq c^\star) = \mathbb{P}\left(\exists c \neq c^\star : [\boldsymbol{\mu}_L]_c > [\boldsymbol{\mu}_L]_{c^\star}\right). \tag{22}$$

Applying the union bound, we can upper-bound this error probability as the sum of pairwise error probabilities:

$$\mathbb{P}(y^t \neq c^\star) \leq \sum_{c \neq c^\star} \mathbb{P}\left([\boldsymbol{\mu}_L]_c - [\boldsymbol{\mu}_L]_{c^\star} > 0\right). \tag{23}$$

For any incorrect class $c \neq c^\star$, define the score difference as $\Delta_c = [\boldsymbol{\mu}_L]_c - [\boldsymbol{\mu}_L]_{c^\star}$. Recall from Lemma 1 that the filtered prediction follows a multivariate Gaussian distribution:

$$\boldsymbol{\mu}_L \sim \mathcal{N}(\mathbf{s} + \bar{\mathbf{b}}, \Sigma_L \mathbf{I}). \tag{24}$$

Since the components of $\boldsymbol{\mu}_L$ are independent given the diagonal covariance matrix, the difference Δ_c is a linear combination of independent Gaussians and is therefore also normally distributed:

$$\Delta_c \sim \mathcal{N}\left([\mathbf{s} + \bar{\mathbf{b}}]_c - [\mathbf{s} + \bar{\mathbf{b}}]_{c^\star}, 2\Sigma_L\right). \tag{25}$$

Substituting the definition of the biased class margin $\Delta_{b,c} = [\mathbf{s} + \bar{\mathbf{b}}]_{c^\star} - [\mathbf{s} + \bar{\mathbf{b}}]_c$, the distribution of Δ_c simplifies to:

$$\Delta_c \sim \mathcal{N}(-\Delta_{b,c}, 2\Sigma_L). \tag{26}$$

The probability that this difference is greater than zero can be evaluated using the standard normal cumulative distribution function, $\Phi(\cdot)$:

$$\mathbb{P}(\Delta_c > 0) = \Phi\left(-\frac{\Delta_{b,c}}{\sqrt{2\Sigma_L}}\right). \tag{27}$$

Applying the standard Chernoff bound for the Gaussian tail, $\Phi(-z) \leq \exp(-z^2/2)$ for $z > 0$, and substituting $z = \frac{\Delta_{b,c}}{\sqrt{2\Sigma_L}}$, we obtain:

$$\Phi\left(-\frac{\Delta_{b,c}}{\sqrt{2\Sigma_L}}\right) \leq \exp\left(-\frac{1}{2}\left(\frac{\Delta_{b,c}}{\sqrt{2\Sigma_L}}\right)^2\right) = \exp\left(-\frac{\Delta_{b,c}^2}{4\Sigma_L}\right). \tag{28}$$

Substituting this pairwise probability back into the union bound yields the error bound:

$$\mathbb{P}(y^t \neq c^\star) \leq \sum_{c \neq c^\star} \exp\left(-\frac{\Delta_{b,c}^2}{4\Sigma_L}\right). \tag{29}$$

Assuming that $\mathbf{b}_l = \mathbf{0}$ for all l, the systematic bias vanishes ($\bar{\mathbf{b}} = \mathbf{0}$). Consequently, the biased class margin $\Delta_{b,c}$ reduces to the clean class margin $\Delta_{0,c} = [\mathbf{s}]_{c^\star} - [\mathbf{s}]_c$. The upper bound thus simplifies to:

$$\mathbb{P}(y^t \neq c^\star) \leq \sum_{c \neq c^\star} \exp\left(-\frac{\Delta_{0,c}^2}{4\Sigma_L}\right). \tag{30}$$

Recall from Lemma 1 that the posterior variance is lower than the minimum individual observation variance, i.e., $\Sigma_L < \sigma_{best}^2$. Because $f(x) = \exp(-A/x)$ is a monotonically increasing function for $A > 0$ and $x > 0$, substituting Σ_L with the larger variance σ_{best}^2 increases the bound:

$$\sum_{c \neq c^\star} \exp\left(-\frac{\Delta_{0,c}^2}{4\Sigma_L}\right) < \sum_{c \neq c^\star} \exp\left(-\frac{\Delta_{0,c}^2}{4\sigma_{best}^2}\right). \tag{31}$$

This establishes that the theoretical misclassification upper bound of the filtered pseudo-label is lower than that of the single best prediction, completing the proof.

A.3 Algorithm

Algorithm 2: The p-MoCaFe Framework

Input: Target training dataset $\mathcal{X}$, public dataset $\mathcal{S}_{\text{v}}$, and pre-trained model f
Output: Adapted model $f^\star$

Execute Line 1 to Line 18 in Algorithm 1 on $\mathcal{X}$ to obtain pseudo-labels $y^t(x)$, filtered logits $\boldsymbol{\mu}(x)$, and subset $\mathcal{X}_m$;
$U \leftarrow \emptyset,\ k \leftarrow \left\lceil \sqrt{|\mathcal{S}_{\text{v}}|}/2 \right\rceil$;
Construct filtered pseudo-labels $y^t(x)$ and logits $\boldsymbol{\mu}(x)$ for sequences in $\mathcal{S}_{\text{v}}$ via Eq. (6);
for $(x, y) \in \mathcal{S}_v$ **do**
    $X \leftarrow (\max_c[\sigma(\boldsymbol{\mu}(x))]_c,\ \max_c[\sigma(f(x))]_c)$;
    if $y^t(x) = y$ **then**
        $(p, d) \leftarrow (0, 1)$;
    else
        $(p, d) \leftarrow \left(1,\ \frac{\ell(y^t(x), f(x))}{\ell(y, f(x))}\right)$;
    end
    $U \leftarrow U \cup \{(X, (p, d))\}$;
end
for $x \in \mathcal{X}_m$ **do**
    $Z \leftarrow (\max_c[\sigma(\boldsymbol{\mu}(x))]_c,\ \max_c[\sigma(f(x))]_c)$;
    $(\hat{p}, \hat{d}) \leftarrow k$-NN$(U, Z)$ using standard Euclidean distance in $\mathbb{R}^2$;
    $w_f(x) \leftarrow \min\left\{1,\ \frac{\beta(x)}{1+\hat{p}(\hat{d}-1)}\right\}$;
end
Adapt model f over $\mathcal{X}_m$ by minimizing $R^w(f)$ via Eq. (12);

A.4 Studying the MSE of a Fixed Prediction

In this section, we study the Mean-Squared-Error (MSE) of a fixed prediction f for an arbitrary object $x \in \mathcal{X}$. In what follows, we theoretically demonstrate the case when the weighting scheme could potentially underperform the standard unweighted approach from a bias-variance perspective. The derivations also provide theoretical justification for projecting the weights to the $[0, 1]$ interval (recall Line 16 of Algorithm 2). Formally, for the MSE of standard unweighted approach $\mathrm{MSE}(x)$ and the MSE of our proposed `p-MoCaFe` scheme $\mathrm{MSE}^w(x)$, the following equations hold

$$\begin{aligned} \mathrm{MSE}(x) &= \mathbb{E}_{y^t(x)|x}\left[\left(\ell\left(y, f(x)\right) - \ell(y^t(x), f(x))\right)^2\right], \\ \mathrm{MSE}^w(x) &= \mathbb{E}_{y^t(x)|x}\left[\left(\ell\left(y, f(x)\right) - w_f(x)\ell(y^t(x), f(x))\right)^2\right]. \end{aligned} \tag{32}$$

We have the following proposition.

Proposition 5. *Let $\ell : \mathbb{R}^C \times \mathbb{R}^C \to \mathbb{R}_+$ be a bounded loss function. Fix $x \in \mathcal{X}$ and a predictor $f : \mathcal{X} \to \mathbb{R}^C$. We have $\mathrm{MSE}(x) < \mathrm{MSE}^w(x)$ if and only if:*

1. $d_f(x) < 1/2$;

2. $p(x) \in \left(0, \frac{1-2\cdot d_f(x)}{(1-d_f(x))^2}\right)$.

Proof: We have:

$$\begin{aligned} \mathrm{MSE}(x) &= \mathbb{E}_{y^t(x)|x}\left[\left(\ell\left(y, f(x)\right) - \ell(y^t(x), f(x))\right)^2\right] \\ &= p(x)\left(\ell\left(y, f(x)\right) - \ell\left(f_{\text{false}}(x), f(x)\right)\right)^2 \\ &= p(x)\ell\left(y, f(x)\right)^2\left(1 - d_f(x)\right)^2 \end{aligned}$$

and

$$\begin{aligned} \mathrm{MSE}^w(x) =& \mathbb{E}_{y^t(x)|x}\left[\left(\ell\left(y, f(x)\right) - w_f(x)\ell(y^t(x), f(x))\right)^2\right] \\ =& (1 - p(x))\ell\left(y, f(x)\right)^2\left(1 - w_f(x)\right)^2 \\ &+ p(x)\left(\ell\left(y, f(x)\right) - w_f(x)\ell\left(f_{\text{false}}(x), f(x)\right)\right)^2 \\ =& (1 - p(x))\ell\left(y, f(x)\right)^2\left(1 - w_f(x)\right)^2 \\ &+ p(x)\ell\left(y, f(x)\right)^2\left(1 - w_f(x)d_f(x)\right)^2. \end{aligned}$$

Proposition 5 implies that when it is less likely for the teacher to corrupt the label of data sample x, i.e., $p(x)$ is sufficiently small, and the prediction of the student is close enough to the corrupted label (i.e., when $d_f(x)$ is small enough), we can use the unweighted estimator instead of the weighted one from a bias-variance trade-off perspective, as the former has smaller MSE in this case. This observation aligns well with our method as we always project $w_f(x)$ to $[0, 1]$ ($w_f(x) > 1$ iff $d_f(x) < 1$ and $p(x) > 0$).

A.5 Proof of Proposition 3

Let $\mathbb{X}$ be an unknown distribution over objects, i.e., $x \sim \mathbb{X}$. Also, each $x \in \mathcal{X}_m$ has a ground-truth label $y \in \mathcal{Y}$. Thus, a clean sample follows $(x, y) \sim \tilde{\mathbb{Q}}_{\text{clean}}$. To quantify the noise induced by filtered pseudo-labels, we consider a stochastic teacher that outputs a "corrupted" label $f_{\text{false}}(x)$ with probability $p(x)$ and the ground-truth label y with probability $1-p(x)$, where $\tilde{\mathbb{Q}}_{\text{t}}$ denotes the induced distribution over data samples and filtered pseudo labels. Considering the weighting strategy in (11), we have

$$
\begin{aligned}
\mathbb{E}\left[R^w(f)\right] &= \mathbb{E}_{\mathcal{X}_m \sim \tilde{\mathbb{Q}}_{\text{t}}}\left[\frac{1}{m}\sum_{x\in\mathcal{X}_m} w_f\left(x\right)\ell\left(y^t(x), f\left(x\right)\right)\right] \\
&= \mathbb{E}_{x\sim\mathbb{X}}\left[\mathbb{E}_{y^t(x)|x}\left[w_f(x)\ell(y^t(x), f(x))\right]\right] \\
&= \mathbb{E}_{x\sim\mathbb{X}}\left[w_f(x)\cdot\left(p(x)\ell\left(f_{\text{false}}(x), f(x)\right) + (1-p(x))\ell\left(y, f(x)\right)\right)\right] \\
&= \mathbb{E}_{x\sim\mathbb{X}}\left[\frac{\beta(x)p(x)\ell\left(f_{\text{false}}(x), f(x)\right) + \beta(x)(1-p(x))\ell\left(y, f(x)\right)}{1+p(x)\cdot(d_f(x)-1)}\right] \\
&= \mathbb{E}_{x\sim\mathbb{X}}\left[\frac{\beta(x)\ell\left(y, f(x)\right) + \beta(x)\ell\left(y, f(x)\right)\cdot p(x)\cdot(d_f(x)-1)}{1+p(x)\cdot(d_f(x)-1)}\right]. \\
&= R(f).
\end{aligned}
\tag{33}
$$

A.6 Discussion on the Estimation of (11)

We provide some discussions on the estimation of (11) and Algorithm 2 for practical implementations.

First, the number k of neighbors for weight estimation is set to $k = \lceil\sqrt{v}/2\rceil$ in Algorithm 2, where v is the size of the public dataset. This is because choosing $k = \alpha\left(v^{2/(2+\dim)}\right)$, where $\dim$ is the dimension of the underlying metric space ($\dim = 2$ in our case), is asymptotically optimal, and $\alpha = 1/2$ is a popular choice for the constant used in practice [44], [45], we also provide the ablation study in Table 9.

Second, notice that (11) indicates that the weight of an example can exceed 1 if and only if the corresponding distortion value in (11) for that example is less than 1. This situation might arise, for instance, when both the student and teacher have the same (or very similar) inaccurate predictions for a given example. In such cases, the weight value in (11) suggests that the risk associated with this example should be higher than the low value implied by the unweighted loss function. However, since ground-truth labels are unavailable during training and only the teacher's inaccurate prediction is accessible, assigning a weight greater than 1 in this scenario could lead to the model overfitting the inaccurate label. To prevent this issue, we project all weights onto the $[0, 1]$ interval. Additionally, Appendix A.4 provides further justification for this projection by analyzing the benefits of constraining weights for low-distortion examples within $[0, 1]$ range using MSE considerations.

At last, according to (11), the weight of an example is a function of the predictor, which, in the case of neural networks, corresponds to the model parameters. This means that ideally we should update our weights assignment every time the parameters get updated during training. However, we empirically observe that even estimating the weight assignment only once during the whole training process is sufficient to achieve our objectives. As a result, our method adds minimal overhead to the standard training process.

A.7 Proof of Proposition 4

Taking the expectation of the clipped weighted empirical risk over the sampling distribution Q and the pseudo-label noise generation process gives:

$$\mathbb{E}[\hat{R}^{\tilde{w}}(f)] = \mathbb{E}_{x\sim Q}\Big[\tilde{w}_f(x)\mathbb{E}_{y^t|x}[\ell(y^t(x), f(x))]\Big]. \tag{34}$$

By the pseudo-label noise model, the inner expectation expands to:

$$\mathbb{E}_{y^t|x}[\ell(y^t(x), f(x))] = (1-p(x))\ell(y, f(x)) + p(x)\ell(f_{\text{false}}(x), f(x)). \tag{35}$$

Substituting the definition of the distortion factor, $\ell(f_{\text{false}}(x), f(x)) = d_f(x)\ell(y, f(x))$, we obtain:

$$\mathbb{E}_{y^t|x}[\ell(y^t(x), f(x))] = \big[1 + p(x)(d_f(x)-1)\big]\ell(y, f(x)). \tag{36}$$

For notational simplicity in the subsequent steps, we define the shorthand $a(x) = p(x)(d_f(x)-1)$ and its empirical estimate $\hat{a}(x) = \hat{p}(x)(\hat{d}_f(x)-1)$. Hence, the expected empirical risk simplifies to:

$$\mathbb{E}[\hat{R}^{\tilde{w}}(f)] = \mathbb{E}_{x\sim Q}\Big[\tilde{w}_f(x)\big[1 + a(x)\big]\ell(y, f(x))\Big]. \tag{37}$$

Concurrently, via importance sampling, the clean risk can be rewritten under Q as:

$$R(f) = \mathbb{E}_{x\sim\Pi}[\ell(y, f(x))] = \mathbb{E}_{x\sim Q}[\beta(x)\ell(y, f(x))]. \tag{38}$$

Since $\hat{w}_f(x) = \frac{\beta(x)}{1+\hat{a}(x)}$, it follows that $\beta(x) = \hat{w}_f(x)\big[1+\hat{a}(x)\big]$. Substituting this into the clean risk formulation yields:

$$R(f) = \mathbb{E}_{x\sim Q}\Big[\hat{w}_f(x)\big[1 + \hat{a}(x)\big]\ell(y, f(x))\Big]. \tag{39}$$

Subtracting the true risk from the expected empirical risk isolates the total deviation:

$$\mathbb{E}[\hat{R}^{\tilde{w}}(f)] - R(f) = \mathbb{E}_{x\sim Q}\Big[\Big(\tilde{w}_f(x)\big[1+a(x)\big] - \hat{w}_f(x)\big[1+\hat{a}(x)\big]\Big)\ell(y, f(x))\Big]. \tag{40}$$

To decouple the k-NN estimation error from the structural clipping effect, we rewrite the inner term by strategically adding and subtracting $\tilde{w}_f(x)\big[1+\hat{a}(x)\big]$:

$$\tilde{w}_f(x)\big[1+a(x)\big] - \hat{w}_f(x)\big[1+\hat{a}(x)\big] = \tilde{w}_f(x)\big(a(x)-\hat{a}(x)\big) + \big(\tilde{w}_f(x)-\hat{w}_f(x)\big)\big[1+\hat{a}(x)\big]. \tag{41}$$

Taking the absolute value of the expectation, applying both Jensen's inequality ($|\mathbb{E}[X]| \le \mathbb{E}[|X|]$) and the triangle inequality ($|A+B| \le |A| + |B|$), and substituting $a(x)$ and $\hat{a}(x)$ back to their original definitions yields the stated upper bound:

$$\begin{aligned}\big|\mathbb{E}[\hat{R}^{\tilde{w}}(f)] - R(f)\big| \le{}& \mathbb{E}_{x\sim Q}\Big[\tilde{w}_f(x)\big|p(x)(d_f(x)-1) - \hat{p}(x)(\hat{d}_f(x)-1)\big|\ell(y, f(x))\Big] \\ &+ \mathbb{E}_{x\sim Q}\Big[\big|\tilde{w}_f(x) - \hat{w}_f(x)\big|\big[1+\hat{p}(x)(\hat{d}_f(x)-1)\big]\ell(y, f(x))\Big].\end{aligned} \tag{42}$$

B Experiments

B.1 Ablation Study

Robustness of Pseudo-label Construction. We provide a comprehensive evaluation of pseudo-label robustness using ResNet-50 in Table 4 and Swin-T in Table 5. In comparison to selecting the highest-confidence (lowest entropy) prediction from each sequence as the teacher (i.e., the max-confidence teacher), the filtered teacher employed in `MoCaFe` demonstrates stronger robustness. In particular, when the pre-trained model is relatively strong (approximately 75% accuracy), the filtered teacher attains performance comparable to that of max-confidence selection. However, as model quality severely deteriorates, the filtered teacher clearly surpasses the max-confidence teacher when the pre-trained models are very poor (around 40%/50% accuracy), a regime in which learning from the max-confidence teacher (which can be confidently wrong) may mislead model adaptation. In contrast, our information filter-based approach mitigates this corruption by aggregating multiple observations and downweighting unreliable views based on cross-view consensus, offering a clear empirical advantage.

Specifically, when using ResNet-50 at 75% initial accuracy, the filtered teacher yields modest gains over max-confidence selection: 0.74 points in pseudo-label correctness and 0.57 points in final test accuracy. However, under severely degraded initializations (49.9% and 42.2%), these final accuracy improvements surge to 6.54 and 3.71 points, respectively. Notably, these gains are substantially larger when employing Swin-T. Our interpretation is that Swin-T has stronger representation and more powerful target-domain adaptation capacity and can therefore exploit improved pseudo-supervision more effectively. For instance, at 75.0% pretrained accuracy, the filtered teacher improves pseudo-label correctness by 2.18 points and test accuracy by 0.79 points over the max-confidence baseline. Under degraded initializations of 57.0% and 51.0%, the final test accuracy improvements expand significantly to 4.57 and 6.01 points, respectively.

Table 4: Robustness of pseudo-label construction. We investigate the pseudo-label construction module in `MoCaFe` with ResNet-50 under the Boston Seaport scenario. "PL correctness" measures the correctness percentage of constructed pseudo-labels.

Pre-train Acc.	Pseudo-labeling Strategy	PL Correctness (%)	Test Acc. (%)
75.0%	Max-confidence	84.21	80.24
	Filtered (proposed)	**84.95**	**80.81**
49.9%	Max-confidence	50.88	46.67
	Filtered (proposed)	**55.13**	**53.21**
42.2%	Max-confidence	44.89	47.32
	Filtered (proposed)	**47.43**	**51.03**

Table 5: Robustness of pseudo-label construction. We investigate the pseudo-label construction module in `MoCaFe` with Swin-T under the Boston Seaport scenario.

Pre-train Acc.	Pseudo-labeling Strategy	PL Correctness (%)	Test Acc. (%)
75.0%	Max-confidence	86.54	88.81
	Filtered (proposed)	**88.72**	**89.60**
57.0%	Max-confidence	64.68	68.23
	Filtered (proposed)	**68.35**	**72.80**
51.0%	Max-confidence	58.80	61.56
	Filtered (proposed)	**63.76**	**67.57**

Effect of Key Modules. We evaluate the impact of our two core components in Table 6. `MoCaFe` yields a significant performance gain, which demonstrates the effectiveness of the data sampling strategy. However, since `MoCaFe` lacks the weighting scheme, it performs slightly worse than

p-MoCaFe. Moreover, applying our weighting scheme alone with random sampling achieves the worst performance. This indicates that data sampling, which constructs effective teacher-student pairs, is an indispensable component of both MoCaFe and p-MoCaFe.

Table 6: Ablation study on key components of p-MoCaFe in the Boston Seaport scenario on the nuScenes dataset with best in **bold**.

Data Sampling	Data Weighting	ResNet-50 75%	ResNet-50 70%	ResNet-50 65%	Swin-T 75%	Swin-T 70%	Swin-T 65%
✗	✔	79.16↑(± 0.18)	75.516↑(± 0.31)	67.04↑(± 0.25)	78.08↑(± 0.22)	74.82↑(± 0.26)	62.76↓(± 0.38)
✔	✗	80.81↑(± 0.23)	77.25↑(± 0.25)	69.32↑(± 0.24)	89.60↑(± 0.11)	89.34↑(± 0.15)	89.27↑(± 0.16)
✔	✔	**81.23↑(± 0.13)**	**77.93↑(± 0.16)**	**69.74↑(± 0.25)**	**90.19↑(± 0.17)**	**90.07↑(± 0.18)**	**90.02↑(± 0.12)**

Effect of Public (Validation) Dataset for p-MoCaFe. In our experiments, we use the source public dataset (Singapore Onenorth in nuScenes) for p-MoCaFe, yet it effectively adapted to *diverse* target domains, e.g., Boston Seaport, KITTI dataset, and the embodied agent scenario which has a large domain gap (results in Tables 1-2 and 11-14 in Appendix B.3-Appendix B.5). For each target task, we align the public labels with the classifier's output categories and retain only compatible samples to estimate the confidence-based calibration quantities. Since the public dataset is used to characterize the relationship between confidence and correctness rather than domain features, our results indicate that p-MoCaFe is *non-sensitive to the domain mismatch* between the public dataset and the target environment. We also provide ablation studies on the choice and size of the public dataset in Tables 7 and 8. When another public dataset is used to calibrate Boston, we apply the same classifier to its object sequences and compute confidence and pseudo-label correctness using only samples with semantically matched categories. Across different public datasets and sizes, p-MoCaFe exhibits small performance variations (< 0.4 percentage points across datasets and < 0.7 percentage points of degradation even at 10% public dataset size) and remains close to MoCaFe.

Table 7: Ablation study on public datasets in p-MoCaFe, where the target domain is Boston Seaport and the pretrained accuracy is 75%.

Validation Set	Singapore	KITTI	Embodied
ResNet-50	81.23↑(± 0.13)	81.16↑(± 0.26)	80.97↑(± 0.25)
Swin-T	90.19↑(± 0.17)	90.06↑(± 0.22)	89.84↑(± 0.24)

Table 8: Ablation study on public dataset size using p-MoCaFe for Boston Seaport. X% denotes the percentage of the original public dataset size.

% of Size	100%	50%	10%
ResNet-50	81.23↑(± 0.13)	80.92↑(± 0.24)	80.71↑(± 0.25)
Swin-T	90.19↑(± 0.17)	89.67↑(± 0.23)	89.52↑(± 0.29)

Effect of Hyperparameter k. For MoCaFe, there is no other framework-induced hyperparameter to adjust across datasets and models. For p-MoCaFe, we show the influence of the neighborhood size k in the k-NN method in Table 9. Our framework maintains very stable performance across different k. The phenomenon shows that both MoCaFe and p-MoCaFe are insensitive to hyperparameters. This insensitivity stems from our design principle, which does not rely on modifications to the learning objective.

Effect of Availability of d. Our algorithm can be implemented regardless of whether d is available. When $d_{i,l}$ is available (e.g., from the radar), it is used only to assign observations to mobility intervals and therefore a coarse range estimate is sufficient (e.g., D_j is set as 0-40m, 40-80m, >80m). We do not need a precise object-to-agent distance. If distance is unavailable, our method remains effective, as shown in Table 10.

Table 9: Ablation study on k using the proposed p-MoCaFe framework in the Boston Seaport scenario with best in **bold**.

Pre-trained	ResNet-50 75%	ResNet-50 70%	ResNet-50 65%
$k = \lceil\sqrt{v}/4\rceil$	81.15 (± 0.16)	77.83 (± 0.18)	69.64 (± 0.20)
$k = \lceil\sqrt{v}/2\rceil$	**81.23 (± 0.13)**	**77.93 (± 0.16)**	**69.74 (± 0.25)**
$k = \lceil\sqrt{v}\rceil$	81.19 (± 0.19)	77.88 (± 0.21)	69.70 (± 0.28)

Table 10: Ablation study on the availability of d using the proposed MoCaFe framework in Boston Seaport.

Pre-trained	ResNet-50 75%	ResNet-50 65%	Swin-T 75%	Swin-T 65%
u only	80.81 (± 0.25)	69.28 (± 0.21)	89.62 (± 0.15)	89.24 (± 0.13)
u & d	80.81 (± 0.23)	69.32 (± 0.24)	89.60 (± 0.11)	89.27 (± 0.16)

B.2 Extension to Object Detection Models

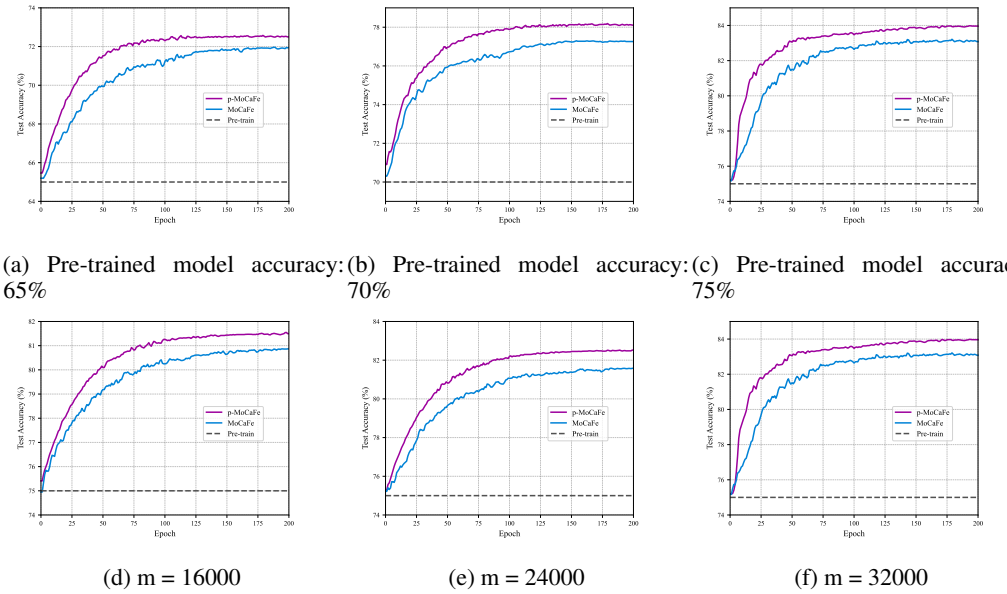
B.2.1 Method

Beyond single-label classification, let us consider extending the `MoCaFe` framework to adapt the classification module in object detection models. For instance, a well-known object detection model, Faster R-CNN, has a dedicated classification network, which can be fine-tuned while keeping other parts frozen (i.e., freezing the backbone and the regression network). The main difference from our previous design is that the model can classify multiple objects within a single frame. To employ `MoCaFe`, we first construct sequences of image frames by their time stamps, where images recorded closely over time are constructed as a sequence. A filtered teacher pseudo-label is then constructed for each individual object track across the sequence. Consequently, we calculate the overall discrepancy score $h(x)$ for an entire image frame as the *summation* of the individual discrepancy scores between each object's prediction and its corresponding teacher output. Essentially, the main distinction from single-label classification lies in this aggregated method used to calculate the discrepancy score. The data weighting scheme in (11) can still be applied to yield an unbiased estimator of the clean loss. In this way, we seamlessly adapt the informative data sampling and debiased weighting procedures from Sections 4.1 and 4.2 to the classification head of object detection models.

B.2.2 Experimental Results for Faster R-CNN

In what follows, we provide experimental results by extending our `MoCaFe` framework to a well-known object detection model, Faster R-CNN with ResNet-50 as the backbone network. Since the TTA baselines in Section 5 focus on single-label classification, which cannot be easily adapted to object detection models, we only evaluate our approaches, including `MoCaFe` and `p-MoCaFe`, while merely comparing ours with existing TTA schemes in previous single-label classification scenarios.

As shown in Fig. 3, `p-MoCaFe` achieves the best test accuracy across varying settings of pre-trained model accuracy and number of data samples. Moreover, `p-MoCaFe` and `MoCaFe` outperform the pre-trained model by a substantial margin (e.g., 9%). This suggests that our method can be applied to improve the classification modules of object detection models without manual labeling, thereby directly benefiting a wide range of practical applications with movable agents, including autonomous driving, drones, and robots.



(a) Pre-trained model accuracy: 65% (b) Pre-trained model accuracy: 70% (c) Pre-trained model accuracy: 75%

(d) m = 16000 (e) m = 24000 (f) m = 32000

Figure 3: The test accuracy versus epoch with the object detection model Faster R-CNN.

B.3 Performance on the Number of Selected Samples on the nuScenes Dataset

Table 11 presents the performance of the ResNet-50 and Swin-T architectures on the nuScenes dataset across varying training sample sizes ($m \in \{32,000, 24,000, 16,000\}$). The results demonstrate that our proposed `MoCaFe` and `p-MoCaFe` frameworks consistently outperform all other evaluated baselines across all sample-size configurations. Notably, even when the available training data is reduced to 16,000 samples, `p-MoCaFe` maintains robust adaptation performance, achieving the highest accuracy among all test-time adaptation methods.

Table 11: Performance in the Boston Seaport scenario on the nuScenes dataset under different numbers of selected training samples m. The pre-trained model accuracy for the Boston scenario is 75%. **Boldface** and Underline highlight the highest and second-highest accuracy, except for the fully-supervised fine-tuning benchmark.

	ResNet-50			**Swin-T**		
Num. Samples	$m = 32000$	$m = 24000$	$m = 16000$	$m = 32000$	$m = 24000$	$m = 16000$
w/o MoCaFe	76.18↑(± 0.28)	75.36↑(± 0.15)	75.31↑(± 0.18)	76.69↑(± 0.26)	76.06↑(± 0.25)	75.87↑(± 0.28)
AdaContrast	77.88↑(± 0.19)	76.93↑(± 0.23)	76.27↑(± 0.28)	76.85↑(± 0.33)	76.24↑(± 0.35)	75.73↑(± 0.37)
PLUE	77.94↑(± 0.12)	76.48↑(± 0.23)	75.84↑(± 0.24)	77.43↑(± 0.24)	76.95↑(± 0.26)	76.51↑(± 0.25)
RoTTA	78.67↑(± 0.21)	77.93↑(± 0.30)	77.22↑(± 0.29)	—	—	—
C-SFDA	78.71↑(± 0.22)	77.54↑(± 0.27)	76.63↑(± 0.30)	77.75↑(± 0.18)	77.36↑(± 0.20)	76.97↑(± 0.22)
UPA	76.87↑(± 0.36)	76.03↑(± 0.35)	75.47↑(± 0.38)	78.24↑(± 0.26)	77.86↑(± 0.27)	77.35↑(± 0.26)
AnCon	79.23↑(± 0.29)	78.71↑(± 0.33)	78.16↑(± 0.32)	80.38↑(± 0.27)	80.02↑(± 0.25)	79.66↑(± 0.28)
MoCaFe	80.81↑(± 0.23)	80.55↑(± 0.18)	80.01↑(± 0.24)	89.60↑(± 0.11)	89.22↑(± 0.18)	88.94↑(± 0.14)
p-MoCaFe	**81.23↑(± 0.13)**	**80.85↑(± 0.18)**	**80.42↑(± 0.26)**	**90.19↑(± 0.17)**	**89.70↑(± 0.14)**	**89.48↑(± 0.25)**
FSL	88.66↑(± 0.12)	87.56↑(± 0.19)	86.83↑(± 0.28)	92.30↑(± 0.06)	91.85↑(± 0.16)	91.36↑(± 0.24)

B.4 Performance on KITTI Dataset

We also evaluate the performance of our framework by adapting the model to the KITTI dataset, a widely used dataset for autonomous driving. To handle the new label space, we freeze the pre-trained backbone and adapt the classification head to KITTI's specific categories. For adaptation on the KITTI dataset, we follow the same instance sequence construction protocol as in other experiments: objects are extracted from images using ground-truth annotations to form multi-viewpoint sequences. For these, we select m samples to form the adaptation dataset $\mathcal{X}_m$ using the proposed MoCaFe/p-MoCaFe schemes. Similarly, by utilizing perfect sequences, we can isolate the effect of our method from errors introduced by imperfect tracking and demonstrate its empirical potential.

Consistent with the Boston Seaport scenario, we evaluate performance by varying the pre-trained model's accuracy (Table 12) and the training sample size m (Table 13). The simulation results demonstrate consistent and substantial performance improvements of the `MoCaFe` framework compared to pre-trained models across diverse settings. Furthermore, the results also reveal that our `MoCaFe` and `p-MoCaFe` substantially outperform all other TTA baselines. It is worth noting that some baselines may achieve even worse performance than the pre-trained model. However, `MoCaFe` and `p-MoCaFe` can always improve model performance after adaptation.

Table 12: Performance on the KITTI dataset under varied pre-trained model accuracy (i.e., 75%, 70%, and 65%). **Boldface** and Underline highlight the highest and second-highest accuracy, except for the fully-supervised fine-tuning benchmark.

	ResNet-50			**Swin-T**		
Pre-train	75%	70%	65%	75%	70%	65%
w/o MoCaFe	76.20↑(± 0.24)	72.14↑(± 0.28)	60.46↓(± 0.43)	75.98↑(± 0.31)	72.24↑(± 0.28)	41.42↓(± 0.38)
AdaContrast	77.03↑(± 0.30)	73.38↑(± 0.25)	60.74↓(± 0.32)	76.32↑(± 0.28)	73.51↑(± 0.30)	59.78↓(± 0.43)
PLUE	77.56↑(± 0.26)	73.05↑(± 0.31)	67.32↑(± 0.35)	76.88↑(± 0.24)	73.40↑(± 0.27)	65.32↑(± 0.34)
RoTTA	78.16↑(± 0.24)	74.51↑(± 0.28)	67.75↑(± 0.39)	—	—	—
C-SFDA	78.00↑(± 0.20)	74.98↑(± 0.32)	67.79↑(± 0.25)	77.15↑(± 0.23)	72.87↑(± 0.24)	67.64↑(± 0.29)
UPA	76.39↑(± 0.37)	73.44↑(± 0.27)	59.77↓(± 0.53)	76.88↑(± 0.19)	72.57↑(± 0.23)	67.19↑(± 0.28)
AnCon	78.46↑(± 0.25)	75.27↑(± 0.45)	68.62↑(± 0.37)	78.74↑(± 0.24)	74.11↑(± 0.27)	69.04↑(± 0.31)
MoCaFe	81.04↑(± 0.20)	77.95↑(± 0.26)	68.80↑(± 0.29)	84.31↑(± 0.15)	83.43↑(± 0.18)	83.14↑(± 0.26)
p-MoCaFe	**81.37↑(± 0.16)**	**78.16↑(± 0.18)**	**69.02↑(± 0.29)**	**84.79↑(± 0.15)**	**83.89↑(± 0.13)**	**83.40↑(± 0.18)**
FSL	86.98↑(± 0.15)	86.12↑(± 0.27)	85.25↑(± 0.19)	88.54↑(± 0.12)	87.69↑(± 0.21)	86.55↑(± 0.33)

Table 13: Performance on the KITTI dataset under different numbers of selected training samples m. The pre-trained model accuracy on the KITTI dataset is 75%.

Num. Samples	ResNet-50 $m = 32000$	ResNet-50 $m = 24000$	ResNet-50 $m = 16000$	Swin-T $m = 32000$	Swin-T $m = 24000$	Swin-T $m = 16000$
w/o MoCaFe	76.20↑(± 0.24)	75.82↑(± 0.34)	75.37↑(± 0.31)	75.98↑(± 0.31)	74.71↓(± 0.23)	74.24↓(± 0.24)
AdaContrast	77.03↑(± 0.30)	76.75↑(± 0.21)	75.89↑(± 0.24)	76.32↑(± 0.28)	75.94↑(± 0.26)	75.23↑(± 0.34)
PLUE	77.56↑(± 0.26)	77.11↑(± 0.25)	76.74↑(± 0.20)	76.88↑(± 0.24)	76.51↑(± 0.22)	75.86↑(± 0.28)
RoTTA	78.16↑(± 0.24)	77.72↑(± 0.26)	77.30↑(± 0.23)	—	—	—
C-SFDA	78.00↑(± 0.20)	77.53↑(± 0.22)	77.13↑(± 0.27)	77.15↑(± 0.23)	76.73↑(± 0.25)	76.20↑(± 0.29)
UPA	76.39↑(± 0.37)	76.01↑(± 0.35)	75.55↑(± 0.38)	76.88↑(± 0.19)	76.30↑(± 0.21)	75.85↑(± 0.27)
AnCon	78.46↑(± 0.25)	78.05↑(± 0.23)	77.68↑(± 0.26)	78.74↑(± 0.24)	78.23↑(± 0.19)	77.88↑(± 0.21)
MoCaFe	81.04↑(± 0.20)	80.83↑(± 0.18)	80.18↑(± 0.30)	84.31↑(± 0.15)	84.01↑(± 0.17)	83.50↑(± 0.21)
p-MoCaFe	**81.37↑(± 0.16)**	**80.96↑(± 0.22)**	**80.59↑(± 0.25)**	**84.79↑(± 0.15)**	**84.40↑(± 0.17)**	**83.91↑(± 0.25)**
FSL	86.98↑(± 0.15)	86.10↑(± 0.24)	85.76↑(± 0.27)	88.54↑(± 0.12)	87.76↑(± 0.18)	87.28↑(± 0.29)

B.5 Performance on the Embodied Agent Scenario

To further demonstrate the broad applicability of our methodology, we add experiments in the embodied agent domain by adapting the pre-trained models from the Singapore Onenorth scenario. To accommodate the different label space, we freeze the backbone and replace the final classifier with an eight-class head, training it on LaSOT object crops (disjoint from adaptation and evaluation sets). We select frames from the widely investigated tracking datasets, i.e., JRDB [36] (collected by robots), MOTS [37], and LaSOT [38] datasets, with classes, including car, pedestrian, robot, drone, basketball, bottle, dog, and bicycle, to form a dataset for robotic scenarios. We selected 800 sequences and extracted over 30k images used for model adaptation and 5k images for testing. It can be seen that our proposed `MoCaFe` framework consistently outperforms the non-adaptive model and the other baselines. These experiments further demonstrate the broad applicability of our method.

Table 14: Performance on the embodied agent scenario under varied pre-trained model accuracy for this scenario (i.e., 75%, 70%, and 65%).

Pre-train	ResNet-50 75%	ResNet-50 70%	ResNet-50 65%	Swin-T 75%	Swin-T 70%	Swin-T 65%
w/o MoCaFe	76.03↑(± 0.28)	71.06↑(± 0.29)	60.07↓(± 0.34)	76.03↑(± 0.28)	71.32↑(± 0.31)	38.47↓(± 0.35)
AdaContrast	76.85↑(± 0.26)	72.63↑(± 0.25)	65.63↑(± 0.29)	76.81↑(± 0.22)	72.04↑(± 0.24)	58.48↓(± 0.27)
PLUE	77.01↑(± 0.25)	72.88↑(± 0.27)	65.75↑(± 0.33)	76.83↑(± 0.19)	72.79↑(± 0.21)	60.27↓(± 0.30)
RoTTA	77.76↑(± 0.18)	74.00↑(± 0.20)	67.36↑(± 0.23)	—	—	—
C-SFDA	77.93↑(± 0.22)	73.66↑(± 0.22)	68.53↑(± 0.31)	77.12↑(± 0.17)	73.02↑(± 0.26)	66.88↑(± 0.28)
UPA	77.57↑(± 0.27)	72.61↑(± 0.32)	62.23↓(± 0.31)	76.62↑(± 0.30)	72.70↑(± 0.29)	66.51↑(± 0.33)
Ancon	78.33↑(± 0.21)	74.01↑(± 0.27)	67.43↑(± 0.30)	78.89↑(± 0.16)	75.05↑(± 0.26)	68.84↑(± 0.25)
MoCaFe	79.73↑(± 0.15)	77.41↑(± 0.20)	69.03↑(± 0.27)	86.98↑(± 0.19)	86.74↑(± 0.24)	86.70↑(± 0.26)
p-MoCaFe	**80.18↑(± 0.17)**	**77.69↑(± 0.23)**	**69.37↑(± 0.24)**	**87.33↑(± 0.17)**	**87.22↑(± 0.18)**	**87.11↑(± 0.23)**
FSL	86.14↑(± 0.15)	85.77↑(± 0.16)	84.94↑(± 0.20)	89.46↑(± 0.10)	89.26↑(± 0.17)	89.21↑(± 0.20)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state clearly the motivation and core idea of this paper, i.e., adapting vision models on movable agents without manual labeling.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide a full set of assumptions and complete (and correct) proof in Section 4 and Appendices.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experiment settings are provided in Section 5 and Appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All datasets used in our evaluation are publicly available and properly cited within the text. We have not opened the code at this stage, but we will open the code and corresponding instructions on GitHub to facilitate reproducibility after paper acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Detailed training and test details can be found in Section 5 and Appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide error bars in our experiments to show statistical significance. See Section 5 and Appendices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The thorough details on computing resources can be found in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper fully conforms with the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the broader impacts in Section 6.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly mentions the publications that provided the datasets in Section 5.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.